UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação

Tese

Exploration of Encoding Time Reduction Solutions for Intra-Frame Prediction of VVC Encoders

Mário Roberto de Freitas Saldanha

Pelotas, 2021

Mário Roberto de Freitas Saldanha

EXPLORATION OF ENCODING TIME REDUCTION SOLUTIONS FOR INTRA-FRAME PREDICTION OF VVC ENCODERS

Tese apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Luciano Agostini
Coorientadores: Prof. Dr. Gustavo Sanchez
Prof. Dr. César Marcon

Pelotas, 2021

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

S162e Saldanha, Mário Roberto de Freitas

Exploration of encoding time reduction solutions for intra-frame prediction of VVC encoders / Mário Roberto de Freitas Saldanha ; Luciano Volcan Agostini, orientador ; César Augusto Missio Marcon, Gustavo Freitas Sanchez, coorientadores. — Pelotas, 2021.

134 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2021.

1. Video coding. 2. VVC. 3. Intra-frame prediction. 4. Encoding time saving. 5. Machine learning. I. Agostini, Luciano Volcan, orient. II. Marcon, César Augusto Missio, coorient. III. Sanchez, Gustavo Freitas, coorient. IV. Título.

CDD : 005

Elaborada por Maria Inez Figueiredo Figas Machado CRB: 10/1612

Mário Roberto de Freitas Saldanha

EXPLORATION OF ENCODING TIME REDUCTION SOLUTIONS FOR INTRA-FRAME PREDICTION OF VVC ENCODERS

Tese aprovada, como requisito parcial, para obtenção do grau de Doutor em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 13 de dezembro de 2021

Banca Examinadora:

Prof. Dr. Luciano Volcan Agostini (orientador)

Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. César Augusto Missio Marcon (coorientador)

Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Gustavo Freitas Sanchez (coorientador)

Doutor em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul.

Prof. Dra. Carla Liberal Pagliari

Doutora em Sistemas de Computação pela Universidade de Essex.

Prof. Dr. Felipe Martin Sampaio

Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Guilherme Ribeiro Corrêa

Doutor em Ciência da Computação pela Universidade de Coimbra.

Prof. Dr. Pedro António Amado de Assunção

Doutor em Sistemas de Computação pela Universidade de Essex.

ACKNOWLEDGMENTS

This Ph.D. was a journey with several challenges related to its development and the COVID-19 pandemic scenario, where the last two years of this Ph.D. were conducted remotely. I thank all the people I have lived with during these four years.

I would like to thank my friends and advisors Luciano Agostini, Gustavo Sanchez, and César Marcon. Your guidance was very important during this journey. You always believed in my capacity and helped without compromising my own decision at the same time. I appreciate you helping me at all times with any personal or professional problems.

I also want to thank my parents, my family, and my girlfriend for always motivating me to study and try to improve more and more. Being a post-graduate student for so long is not always easy and during this journey we faced happy and difficult moments, but you always supported me and help me to face the challenges.

A big thank you to my friends and colleagues in the GACI/ViTech lab. Thank you for making the lab hours much more fun and for always being available for any kind of conversation.

I want to express my gratitude to Carla, Guilherme, Felipe, and Pedro for being part of the Ph.D. committee and taking the time to review this Thesis.

Finally, thank you to CAPES, FAPERGS, and CNPq, and all the funding agencies that supported my studies.

Thank you all!

“*You may never know what results come of your action, but if you do nothing there will be no result.*” (Mahatma Gandhi)

ABSTRACT

Saldanha, Mário Roberto de Freitas. **Exploration of Encoding Time Reduction Solutions for Intra-Frame Prediction of VVC Encoders**. Advisor: Luciano Agostini 2021. 134f. Thesis (Doctorate in Computer Science) – Graduate Program in Computing, Technology Development Center, Federal University of Pelotas, Pelotas, 2021.

The Versatile Video Coding (VVC) standard was developed to supply the current demand for the upcoming video applications, enabling higher compression performance than previous video coding standards and high versatility to adapt to different types of applications and videos. VVC brings several innovations and enhancements in the coding modules, including larger block sizes, more flexible block partitioning, more angular intra-frame prediction modes, affine motion compensation, and multiple transform selection. Although VVC can provide a high compression performance, these new tools significantly increased the encoding effort. This Thesis proposes timesaving solutions to reduce the encoding effort of the VVC intra-frame prediction. A performance analysis of VVC intra-frame prediction is firstly presented to identify the most time-consuming modules that must be prioritized to accomplish the objective of this work. Subsequently, timesaving solutions based on heuristic and machine learning approaches are presented regarding different intra-frame prediction steps of VVC encoding, including block partitioning of luminance and chrominance samples, mode selection of luminance samples, and the transform selection of luminance samples. It was possible to achieve between 8.5% and 61.3% of encoding time reduction with Bjontegaard Delta Bit Rate (BDBR) ranging from 0.4% to 2.4%. These solutions presented competitive results when compared to the related works.

Keywords: Video Coding, VVC, Intra-Frame Prediction, Encoding Time Saving, Machine Learning.

RESUMO

Saldanha, Mário Roberto de Freitas. **Exploração de Soluções de Redução de Tempo de Codificação para a Predição Intra-Quadro do Codificador VVC**. Orientador: Luciano Agostini. 2021. 134f. Tese (Doutorado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2021.

O padrão de Codificação de Vídeo Versátil, do inglês *Versatile Video Coding* (VVC), foi desenvolvido para atender a demanda atual relacionada às aplicações de vídeo, fornecendo maior eficiência de compressão quando comparado a padrões anteriores de codificação e alta versatilidade para se adaptar a diferentes tipos de aplicações e vídeos. O VVC insere várias inovações e melhoramentos nos módulos de codificação, incluindo blocos de tamanhos maiores, particionamento de bloco mais flexível, mais modos angulares para a predição intra-quadro, compensação de movimento afim, e seleção múltipla de transformadas. Embora o VVC seja capaz de fornecer alta eficiência de compressão, essas novas ferramentas aumentaram o esforço de codificação significativamente. Esta Tese propõe soluções eficientes para reduzir o esforço de codificação da predição intra-quadro do VVC. Uma análise de performance da predição intra-quadro do VVC é apresentada inicialmente para identificar os módulos que consomem mais tempo e que devem ser priorizadas para alcançar o objetivo do trabalho. Posteriormente, soluções para redução do tempo de codificação baseadas em heurísticas e em aprendizado de máquina são apresentadas, considerando diferentes etapas da predição intra-quadro do codificador VVC, incluindo o particionamento de bloco das amostras de luminância e de crominância, a seleção de modo de amostras de luminância e a seleção de transformada de amostras de luminância. Foi possível alcançar entre 8,5% e 61,3% de redução no tempo de codificação com o Bjontegaard Delta Bit Rate (BDBR) variando de 0,4% a 2,4%. Essas soluções apresentaram resultados competitivos quando comparados aos trabalhos relacionados.

Palavras-chave: Codificação de vídeos, VVC, Predição Intra-Quadro, Redução do Tempo de Codificação, Aprendizado de Máquina.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	All-Intra
AIP	Angular Intra Prediction
ALF	Adaptive Loop Filter
AMC	Affine Motion Compensation
AMVP	Advanced Motion Vector Prediction
AMVR	Adaptive Motion Vector Resolution
AR	Augmented Reality
AVC	Advanced Video Coding
AV1	AOMedia Video 1
AVS2	Audio Video Coding Standard 2
BCW	Bi-prediction with CU-level Weights
BDBR	Bjontegaard Delta Bit Rate
BDOF	Bi-directional Optical Flow
BT	Binary Tree
BTH	Binary Tree Horizontal
BTV	Binary Tree Vertical
CABAC	Context-based Adaptive Binary Arithmetic Coding
Cb	Blue Chrominance
CBF	Coded Block Flag
CCLM	Cross-Component Linear Model
C-ETS	Chrominance Encoding Time Saving
CIIP	Combined Inter/Intra Prediction
CNN	Convolutional Neural Network
Cr	Red Chrominance
CSETL	Chrominance CU Splitting Early Termination based on Luminance QTMT
CST	Chroma Separate Tree
CTC	Common Test Conditions
CTU	Coding Tree Unit
CU	Coding Unit
DCT	Discrete Cosine Transform
DF	Deblocking Filter
DM	Derived Mode
DMVR	Decoder-side Motion Vector Refinement
DQ	Dependent Quantization
DST	Discrete Sine Transform
DT	Dual-Tree
EC	Entropy Coding
EFB	Exclusive Feature Bundling
ETS	Encoding Time Saving
FCSDV	Fast Chrominance Split Decision based on Variance of Sub-blocks
FDIS	Final Draft International Standard
FD-ISP	Fast Decision based on ISP
FDV	Fast Decision based on Variance
FHD	Full High Definition
FPS	Frame per Second
GBM	Gradient Boosting Machine

GOP	Group of Pictures
GOSS	Gradient-based One-Side Sampling
GPM	Geometric Partitioning Mode
GPU	Graphics Processing Unit
HD	High Definition
HDR	High Dynamic Range
HEVC	High Efficiency Video Coding
HM	HEVC Test Model
HMVP	History-based Motion Vector Prediction
HSV	Hue, Saturation and Value
IG	Information Gain
ISP	Intra Sub-Partition
JCCR	Joint Coding of Chroma residuals
JCT-VC	Joint Collaborative Team on Video Coding
JVET	Joint Video Experts Team
LD	Low Delay
LFNST	Low-frequency Non-Separable Transform
LGBM	Light Gradient Boosting Machine
LMCS	Luma Mapping with Chroma Scaling
MCL	Merge Candidates List
MC-IF	Media Coding Industry Forum
MIP	Matrix-based Intra Prediction
MMVD	Merge mode with Motion Vector Differences
MPEG	Moving Picture Experts Group
MPM	Most Probable Mode
MR	Mixed Reality
MRL	Multiple Reference Line
MTS	Multiple Transform Selection
MTT	Multi-Type Tree
MV	Motion Vector
MV-HEVC	Multiview HEVC
MVD	Motion Vector Difference
PDPC	Position Dependent Prediction Combination
PU	Prediction Unit
Q	Quantization
QP	Quantization Parameter
QT	Quadtree
QTMT	Quadtree with nested Multi-Type Tree
RA	Random-Access
RD	Rate Distortion
RDO	Rate Distortion Optimization
RF	Random Forest
RGB	Red, Green and Blue
RMD	Rough Mode Decision
SAO	Sample Adaptive Offset
SATD	Sum of Absolute Transformed Differences
SBT	Sub-Block Transform
SDR	Standard Dynamic Range
T	Transform
T-ETS	Total Encoding Time Saving

TPE	Tree-structured Parzen Estimator
TQ	Transform and Quantization
TS	Time Saving
TSM	Transform Skip Mode
TT	Ternary Tree
TTH	Ternary Tree Horizontal
TTV	Ternary Tree Vertical
TU	Transform Unit
U	Blue Chrominance
UHD	Ultra-High Definition
V	Red Chrominance
VCEG	Video Coding Experts Group
VVC	Versatile Video Coding
VTM	VVC Test Model
VR	Virtual Reality
WAIP	Wide-Angle Intra Prediction
WEKA	Waikato Environment for Knowledge Analysis
WPP	Wavefront Parallel Processing
Y	Luminance

CONTENTS

1. INTRODUCTION

The development and popularization of new devices capable of handling digital videos continue to grow, mainly due to the interest of users for high video quality and experiences with immersive technologies such as 3D video, 360º video, Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR) (SHERMAN; CRAIG, 2018). Over the past few years, industry and academia have dedicated efforts to develop devices capable of efficiently handling different types of video content. Currently, users can access a wide variety of these devices, like TVs, notebooks, smartphones, tablets, video games, 360º cameras, and VR glasses. At the same time, advances in telecommunication infrastructures increasing the internet transmission capacity and speed have enabled higher information sharing between devices and, consequently, increased the popularization of digital videos through different services, including Netflix, Amazon Prime Video, YouTube, Instagram, Facebook, and WhatsApp. However, this continuous increase in digital video traffic on the internet has created enormous pressure for increased bandwidth (CISCO, 2020). This became even more noticeable in the COVID-19 pandemic scenario, where the multimedia content consumption over the internet increased a lot, forcing streaming providers to reduce the video quality to support the current demand (DHAPOLA, 2020).

To support this demand for video transmission and storage with increasingly higher resolutions and immersive technologies, it is necessary to develop new advanced techniques for video encoding and decoding systems. There are many challenges in developing these systems, such as reducing the size of the encoded video (bitstream), providing high video quality, achieving real-time processing, achieving low energy consumption, using computational resources according to the environment availability, among others. Therefore, a great effort by industry and academic research groups has been carried out in recent years to define advanced video coding standards, such as Advanced Video Coding (H.264/AVC) (MARPE; WIEGAND; SULLIVAN, 2006), High Efficiency Video Coding (HEVC) (SULLIVAN et al., 2012), VP9 (MUKHERJEE et al., 2013), Audio Video Coding Standard 2 (AVS2) (HE et al., 2013), AOMedia Video 1 (AV1) (CHEN et al., 2018), and Versatile Video Coding (VVC) (BROSS et al., 2021).

Although H.264/AVC has been widely adopted by consolidated platforms, such as Netflix and YouTube, the emergence of Full-High Definition (FHD) and Ultra-High Definition (UHD) video content, immersive technologies, and advanced computer architectures for parallel processing demonstrated that the H.264/AVC did not present enough performance to meet this demand. For this purpose, Joint Collaborative Team on Video Coding (JCT-VC) developed the HEVC video coding standard to increase the coding efficiency over the H.264/AVC and support the growing demand for higher resolutions and immersive technologies. The HEVC was released in 2013, providing the double of compression rate for the same video quality compared to the previous standard (SULLIVAN et al., 2012). This improvement in the coding efficiency comes at the cost of encoding complexity increases of 3.2 and 1.2 times compared to H.264/AVC in All-Intra (AI) and Random-Access (RA) configurations, respectively (VANNE et al., 2012). Besides, HEVC followed the technological advances in the industry with the multicore processors and Graphics Processing Unit (GPU), introducing Tiles and Wavefront Parallel Processing (WPP) (SULLIVAN et al., 2012) to allow an efficient parallelism exploration in the encoding and decoding systems. With the growing demand for immersive technologies that provide experiences beyond 2D videos, JCT-VC has also developed HEVC extensions, such as MV-HEVC and 3D-HEVC (SULLIVAN et al., 2013).

Forecasts by Cisco (2020) indicate that video resolutions, new immersive technologies, and internet video traffic will continue to increase significantly over the next years. Thus, HEVC encoders and decoders are no longer sufficient to meet the market requirements. For this purpose, Joint Video Experts Team (JVET) was created in a collaboration between ISO Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) to develop the VVC standard, which was established as a Final Draft International Standard (FDIS) in July 2020. JVET has designed VVC with a focus on specifying a video coding technology with compression efficiency significantly higher than HEVC and having high versatility for efficient use in a broad range of applications and different types of video content, including UHD, High-Dynamic Range (HDR), screen content, 360º video, and resolution adaptivity. Moreover, to overcome the inhibited industry adoption of HEVC due to the lack of reliable and consolidated licensing structure, a new body called Media Coding Industry Forum (MC-IF) (SAMUELSSON, 2020) was established to define a clear and reasonable licensing model for VVC. Additionally, the first version of VVC already brings several new coding tools to deal with different video applications and contents beyond the standard- and high-definition camera-captured content coding.

VVC introduces several novel techniques and enhancements for block partitioning, intra- and inter-frame prediction, transform, quantization, entropy coding, and in-loop filters to improve the coding efficiency. These improvements include larger block sizes, flexible block partitioning using a Quadtree with nested Multi-Type Tree (QTMT) structure (HUANG et al., 2020, 2021), a higher number of angular intra-frame prediction modes (PFAFF et al., 2021), Affine Motion Compensation (AMC) (ZHANG et al., 2019a), Multiple Transform Selection (MTS) (ZHAO et al. 2016, 2021), Luma Mapping with Chroma Scaling (LMCS) (LU, 2020), extended maximum Quantization Parameter (QP), improved implementations of quantization and entropy coding (CHEN; YE; KIM, 2020).

These new tools improved the encoding efficiency but raised the encoder complexity expressively. Our analysis presented in (SALDANHA et al., 2020a) and the work of Bossen et al. (2021) showed that the VVC Test Model (VTM) (CHEN; YE; KIM, 2020) demands about 27 and 8 times more computational effort than the HEVC Test Model (HM) (ROSEWARNE et al., 2015) for AI and RA encoder configurations, respectively. VVC is a video coding standard that tends to be widely adopted by the industry to meet future market requirements; however, this encoder demands a high computational effort and efficient simplifications are needed to enable real-time processing with low energy consumption.

1.1. Research Question and Research Hypothesis

VVC outperforms HEVC in terms of coding efficiency by 50% for the same video quality at a high computational cost (BROSS et al., 2021). The substantial increase of processing time becomes a challenge to the real-time implementation; it is necessary to work on efficient algorithmic solutions to reduce the VVC encoder complexity while preserving the coding efficiency provided by the new coding tools.

There are several works in the literature proposing solutions to reduce the HEVC encoding complexity. Some of these works propose solutions to reduce the processing time of the block partitioning decision (XU et al., 2018; LIU et al., 2019; CORREA et al., 2019), whereas the works (KIM et al., 2014; GAO et al., 2015) present solutions to reduce the processing time of intra- and/or inter-frame prediction mode decisions. Furthermore, other works focus on the complexity control of the HEVC encoding process (CORREA et al., 2015; DENG; XU, 2017).

Although these solutions present outstanding results in the HEVC encoding process, they cannot be used directly in VVC since it introduces several new tools and modifications in the coding structure, significantly changing the encoding process and context compared to HEVC. Several of these changes have been made to intra-frame prediction, which is the focus of this work, where the computational effort increases by 27 times over HEVC (SALDANHA et al., 2020a). Based on this fact, we define our main research question as:

How to create efficient solutions to reduce the VVC encoding time in the intra-frame prediction while maintaining the coding efficiency?

This question is the base for defining the research hypothesis as:

Exploring heuristic and machine learning techniques as potential approaches for providing impressive encoding time reduction in VVC intra-frame prediction with minimal impact on coding efficiency.

Considering the developed investigation and the reached results, we define the thesis of this work as:

It is possible to reduce encoding time significantly with negligible coding efficiency losses for the novel VVC intra coding tools, exploring heuristics and machine learning techniques.

1.2. Main Objective and Specific Goals

The main objective of this Thesis is to investigate and propose new algorithmic solutions to reduce the encoding time of VVC intra-frame prediction. Figure 1 presents a high-level diagram of the VVC encoder and the Thesis contributions, indicating the chapters where the contributions are detailed according to each coding module.

This Thesis lies in exploring new algorithms and specialized techniques for the design of efficient solutions for reducing the encoding time of VVC intra-frame prediction at several levels of the encoding process. The approaches used are mainly focused on heuristics based on statistical analysis and machine learning techniques. Applying these approaches based on an in-depth analysis of the computational effort and the usage distribution of each encoding mode allowed us to develop efficient solutions and obtain significant results of timesaving with low impact on the coding efficiency.

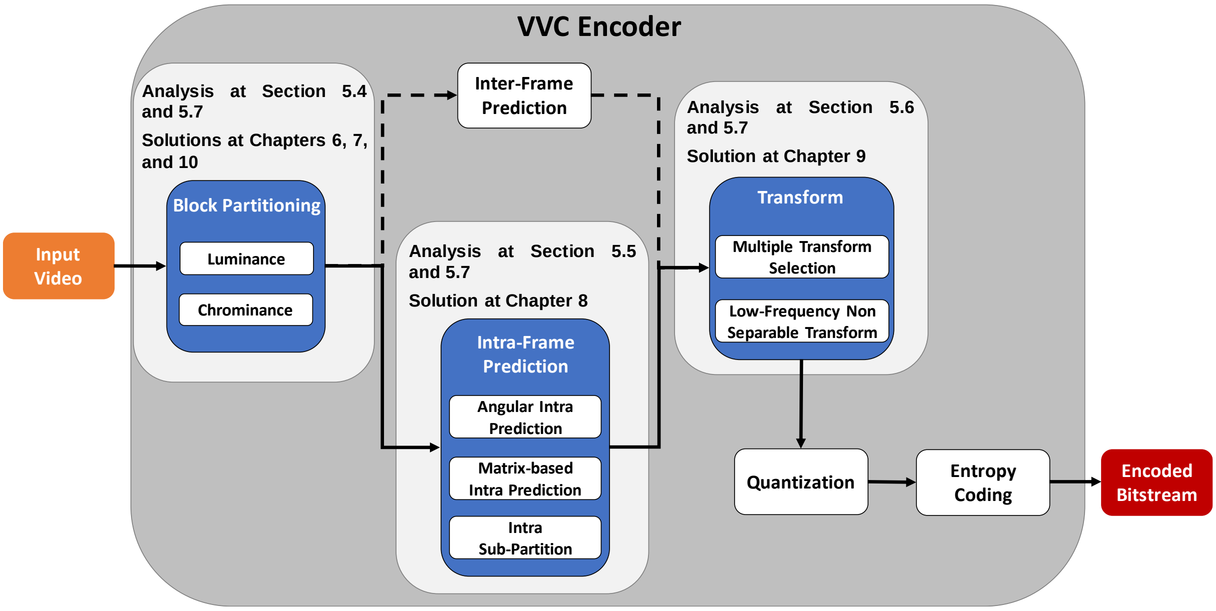


Figure 1 – High-level diagram of the VVC encoder with the Thesis contributions.

To achieve the main objective of this Thesis the following specific goals were accomplished:

- Analyze the processing time and the usage distribution of each block size and coding mode (SALDANHA et al., 2020a, 2021a, 2021e).
- Create fast decision algorithms to speed up the block partitioning selection (SALDANHA et al., 2020b, 2021b, 2021d).
- Create fast decision algorithms to speed up the intra-frame prediction mode selection (SALDANHA et al., 2021c).
- Create fast decision algorithms to speed up the intra-frame prediction transform selection.

1.3. Outline

This Thesis is organized into eleven chapters to present the VVC encoding concepts and contributions. Chapter 2 presents a theoretical background about VVC, discussing some basic aspects of video coding and the main novelties in VVC. Chapter 3 presents the VVC intra-frame prediction, detailing the encoding flow, the novel prediction, and transform coding tools. Chapter 4 presents and discusses a dense set of related works. Chapter 5 presents an extensive performance analysis of VVC intra-frame prediction, showing encoding time and usage distribution analysis. Chapters 6 to 10 present the developed solutions for encoding time reduction and this structure was oriented towards the main publications generated during this doctorate. Chapter 6 presents a fast decision scheme based on heuristic for the block partitioning of luminance blocks. A configurable fast block partitioning solution based on machine learning for luminance blocks is presented in Chapter 7. Chapters 8 and 9 present fast decision schemes for intra mode selection and transform mode selection based on machine learning for luminance blocks, respectively. Chapter 10 presents a fast block partitioning scheme based on statistical analysis for chrominance blocks. Finally, Chapter 11 concludes this Thesis.

2. VERSATILE VIDEO CODING

This chapter describes the VVC theoretical background employed in this Thesis. Section 2.1 introduces some basic concepts about video coding, which are necessary for a complete understanding of this work, and Section 2.2 describes the general aspects of VVC encoding structure, the novel coding tools, and the main innovations of each coding module.

2.1. Basic Concepts

A video sequence consists of a series of static images (commonly called frames) presented sequentially to the viewer at a given temporal rate, and at least 24 frames per second (fps) is required to achieve a smooth motion perception (WIEN, 2015). However, recent video applications have introduced more demanding requirements, increasing the need for higher frame rates. Generally, frame rates of up to 60 fps and 120 fps are specified for High Definition (HD) and UHD videos, respectively (WIEN, 2015). Besides, even higher frame rates are considered for more immersive video (SALMON et al., 2011).

Each frame is represented by a two-dimensional matrix of pixels with horizontal and vertical dimensions, which defines the spatial resolution. The spatial resolution of a video sequence can assume arbitrary values, but some predefined formats are often adopted by industry and supported on many devices, such as 1280×720 (720p), 1920×1080 (1080p), and 3840×2160 (2160p). The perceived video visual quality is highly related to the number of pixels in the image; consequently, the higher the spatial resolution, the better tends to be the visual quality.

Each pixel stores the information of color and luminosity of its corresponding position within each frame. There are several color spaces such as RGB (red, green, and blue), HSV (hue, saturation, and value), and YCbCr. The color space YCbCr is frequently referred to as YUV and denotes luminance (Y), blue chrominance (Cb or U), and red chrominance (Cr or V). In video coding applications, the YCbCr color space is widely adopted in order to take advantage of the lower resolution capability of the human visual system for color (chrominance) with respect to luminosity. This color space enables handling the luminosity information separately from the color information. Moreover, this color space allows performing a subsampling of chrominance information, which is commonly applied in consumer applications (WIEN, 2015).

The subsampling of chrominance components can be considered a kind of video compression since it discards part of the video data without perceptible visual degradation. The most common chrominance subsampling configurations adopted are 4:2:2 and 4:2:0. In the 4:2:2 configuration, there are two Cb and two Cr samples for each four Y samples, whereas, in the 4:2:0 configuration, there is one Cb and one Cr sample for each four Y samples. VVC supports 4:2:2 and 4:2:0 subsampling configurations beyond 4:4:4 and 4:0:0. In the 4:4:4 configuration, no subsampling is applied, whereas, in 4:0:0, only the luminance channel is available. Since the chrominance subsampling configuration 4:2:0 is widely used for several video applications and for most of the work in the literature, we have considered 4:2:0 subsampling configuration along all experiments carried out in this Thesis.

Video coding aims to reduce the amount of redundant data in the representation of the video information, exploring spatial redundancy, temporal redundancy, and entropic redundancy (RICHARDSON, 2010). The spatial redundancy refers to the correlations between samples in the same frame of a video sequence and, in this case, the intra-frame prediction is applied (LAINEMA et al., 2012). The redundancies between neighboring frames are encoded by inter-frame prediction, which explores the correlation of a current frame with previously encoded frames (GHANBARI, 2003). The entropic redundancy refers to the probabilities of occurrences of the coded symbols. In this case, the entropy coding assigns a smaller number of bits to represent symbols with higher occurrence and a larger number of bits to represent symbols with lower occurrence. Current video encoders also include in-loop filters to increase the subjective visual quality (WIEN, 2015).

2.2. Encoding Structure and Novel Coding Tools

VVC follows a block-based hybrid video coding approach, an underlying concept of all major video coding standards, such as H.264/AVC and HEVC. This approach splits each video sequence frame into blocks and processes all blocks sequentially by intra- or inter-frame prediction, transform, quantization, and entropy coding. Figure 2 presents the general encoder structure, demonstrating the data flow at the encoder from the input video sequence to the generated output bitstream.

For each block, a prediction process is performed where either intra- or inter-frame prediction can be applied. The intra-frame prediction explores spatial redundancies, where only information within the frame being encoded is used to explore the correlations between the blocks. The inter-frame prediction explores information present in frames temporally close to the frame being encoded to reduce temporal redundancies. To determine which prediction process to apply for each block, the encoder has an internal decision mode that considers the distortion between the predicted and the original block and the number of bits to represent the prediction.

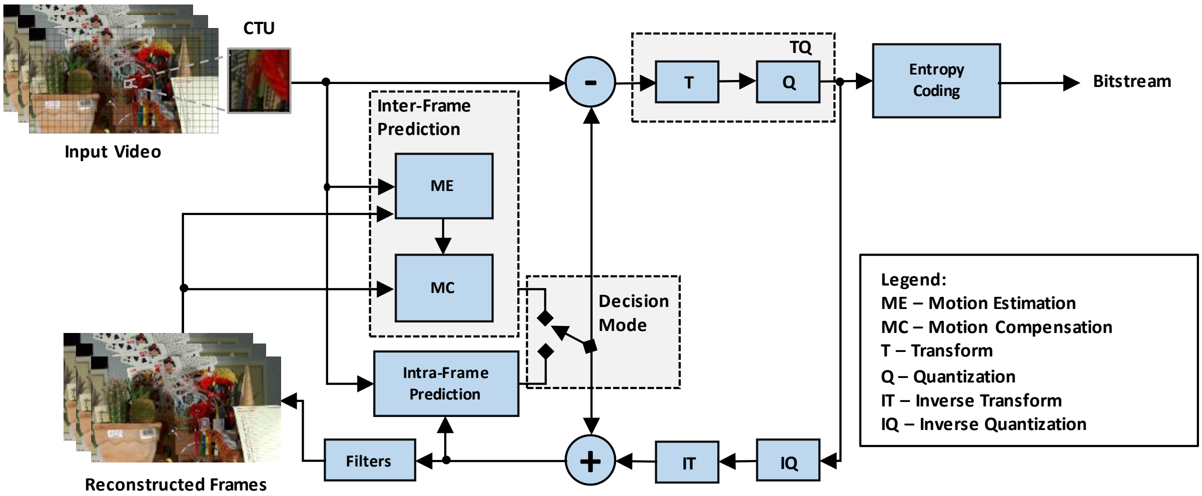


Figure 2 – General encoder structure.

The prediction step generates residues computed from the difference between the predicted and the original block. These residues are processed by the transform module, which converts the values from the spatial domain to the frequency domain. Thus, the quantization step is applied to the transformed values, exploring the characteristics of the human visual system to discard the frequencies that are less relevant to human perception (WIEN, 2015). The transform and quantization steps are called residual coding. After the residual coding, the entropy coding processes the quantized values exploring occurrence probabilities; the values with a higher probability of occurrence are stored with smaller codes in the bitstream.

Since the encoder needs to have the same reference samples as the decoder, the encoder has a reconstruction loop operating as a decoder inside the encoder, ensuring that intra- and inter-frame prediction at the encoder uses the same sample values as the decoder. Since the prediction steps use the samples (or frames) that have already been encoded as references for the next encoding blocks and the quantization is a lossy process, the output of residual coding is also sent to the inverse quantization and inverse transform steps to perform the reconstruction of the residual values. Afterward, the residual values are added to the predicted samples and the reconstructed samples are used as references for intra-frame prediction. For the reconstructed frames, used as references in inter-frame prediction, some in-loop filters are applied to smooth the artifacts inherent to the coding process (WIEN, 2015). Typically, these filters are applied in the decoding system to provide higher visual quality, and, after the filtering process, these frames are used as references in the decoding process. Therefore, the encoder should also apply these filters to have the same reference in the encoding process.

Modern video encoders use a complex Rate-Distortion Optimization (RDO) process to select the best combination of block partitioning and prediction mode. The encoding possibilities are evaluated to select the combination that provides the lowest Rate-Distortion cost (RD-cost). The RD-cost is calculated based on the distortion between the predicted and the original block and the bit rate required for prediction (SULLIVAN; WIEGAND, 1998).

Each video sequence can be divided into a group of frames for the prediction process, known as Group of Pictures (GOP). A GOP is a set of encoded frames containing all the information needed to decode within itself; a GOP is independent of the frames that do not belong to it. All GOPs have the same size, which is statically defined by an encoder configuration parameter.

A GOP can contain three types of frames: (i) I-frame, (ii) P-frame, and (iii) B-frame. I-frames are encoded using only intra-frame prediction. P-frames and B-frames can use both intra- and inter-frame prediction. P-frames which in addition to intra-frame prediction also allows inter-frame prediction from one reference frame per block using one motion vector and one reference index. Finally, B-frames which in addition to intra and uni-prediction also allows inter-frame prediction using two motion vectors and two reference indices (SULLIVAN et al., 2012).

In VVC, each video sequence frame can be divided into slices, which represent a frame region that can be independently decoded from other regions in the same frame. Each slice groups sequences of Coding Tree Units (CTUs), which can be divided into Coding Units (CUs). These structures will be better discussed later. Each CU in a slice is encoded according to a predefined slice type, which can be I, P, or B. In I-slices, all CUs are encoded using only intra-frame prediction. In P-slices, in addition to intra-frame prediction, CUs can also be encoded with unidirectional inter-frame prediction. In B-slices, in addition to intra-frame prediction and unidirectional inter-frame prediction, CUs can also be encoded with bidirectional inter-frame prediction.

Slices are used to avoid significant losses of the encoded data in case of transmission errors of the bitstream and are also used as a strategy for parallel processing. VVC also allows other types of frame divisions used for different applications, such as Tiles, Wavefront, and Subpictures (BROSS et al., 2021), but that are outside the scope of this work. More details can be found in (CHEN; YE; KIM, 2020).

Although VVC follows a similar coding structure to HEVC, it introduces several novel techniques and enhancements for block partitioning, intra- and inter-frame prediction, transform, quantization, entropy, and in-loop filters to improve the encoding efficiency.

2.2.1. Block Partitioning

Since block partitioning plays an essential role in the compression efficiency of modern video encoders, several schemes of block partitioning structure were investigated for VVC. Currently, VVC supports block sizes larger than HEVC to provide an efficient compression rate, especially for high and ultra-high video resolutions. The VVC standard splits each input frame into CTUs covering square regions of at most 128×128 samples. Moreover, each CTU is further recursively partitioned into smaller blocks referred to as CUs.

VVC adopts a coding-tree-based splitting process that, in addition to the HEVC Quadtree (QT) structure, introduces the Multi-Type Tree (MTT) partitioning structure, enabling rectangular CU shapes through Binary Tree (BT) and Ternary Tree (TT). These combined structures (QT+MTT) are named Quadtree with nested Multi-Type Tree (QTMT) and allow six types of partitions shown in Figure 3. A CU can be defined as no split, and the coding process is carried out with the current CU size. Otherwise, a CU can be split with QT, BT, and TT structures. QT splits a CU into four equal-sized square sub-CUs. BT divides horizontally (BTH) or vertically (BTV) a CU into two symmetric sub-CUs. TT splits a CU into three sub-CUs with the ratio of 1:2:1, and the division also can be performed in horizontal (TTH) and vertical (TTV) directions.

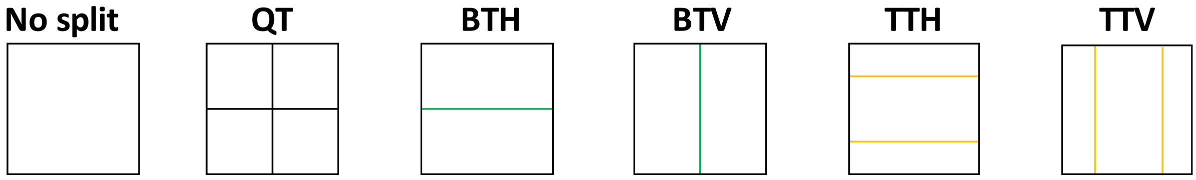


Figure 3 – Partition types of the QTMT partitioning structure.

A CTU is firstly recursively partitioned with a QT structure. Subsequently, each QT leaf node can be further recursively partitioned with an MTT structure using binary and ternary splits. However, when an MTT split is performed, a QT split is no longer allowed. The CU sizes may vary from 4×4 samples up to 128×128 (maximum CTU size) with a total of 28 block sizes from the CTU, including square and rectangular shapes. The QT and MTT leaf nodes represent the CUs, which also are the units used for prediction and transformation. Therefore, the VVC partitioning structure removes the separation among CU, Prediction Unit (PU), and Transform Unit (TU) concepts used in HEVC, where each unit may have a different block size (CHEN; YE; KIM, 2020). An exception occurs when the CU size is larger than the maximum transform length and it is automatically split in the horizontal and/or vertical direction to meet the transform size restriction in that direction (CHEN; YE; KIM, 2020).

The CTU partitioning for luminance and chrominance can be jointly or independently, referred to as single-tree or Dual-Tree (DT) – also known as Chroma Separate Tree (CST), respectively. The single-tree is employed for P- and B-slices, where both intra- and inter-frame predictions can be applied, whereas the dual-tree is used for I-slices, where only the intra-frame prediction can be performed. More details about the partitioning process for intra-frame prediction are provided in Chapter 3.

The QTMT partitioning structure enables high flexibility to represent the block sizes and shapes. Thus, these block partition types can adapt to various video characteristics resulting in better coding efficiency. However, this high flexibility also results in a more computational effort since each CU needs to be evaluated using the RDO process to select the optimal CU partitioning. This evaluation is performed recursively with all splitting possibilities structure, including no split, QT, BTH, BTV, TTH, and TTV.

2.2.2. Prediction Tools

VVC also introduces several modifications for intra- and inter-frame prediction, aiming to improve the encoding performance further. For this purpose, enhancements in the HEVC-based coding tools and novel approaches were adopted. This section briefly describes these improvements, and an extensive discussion of the intra-frame prediction, which is the focus of this Thesis, is provided in Chapter 3.

VVC intra-frame prediction brings several innovations, mainly for the luminance channel. Regarding the conventional intra-frame prediction approach used in HEVC, VVC uses the Planar and DC modes like HEVC and provides finer-granularity angular prediction, extending the HEVC angular prediction modes from 33 to 65 angular modes. VVC enables the intra-frame prediction for rectangular block shapes using the Wide-Angle Intra Prediction (WAIP) approach (ZHAO et al., 2019). Additionally, Position Dependent Prediction Combination (PDPC) (PFAFF et al., 2021) and two types of 4-tap interpolation and smoothing filters are used in the prediction process to reduce the prediction error and to improve the coding efficiency. Multiple Reference Line (MRL) (CHANG et al., 2019) allows more reference lines for VVC intra-frame prediction.

Matrix-based Intra Prediction (MIP) (SCHAFER et al., 2019) is a novel prediction tool designed by a learning-based method and is an alternative approach to the conventional angular intra-frame prediction modes. Intra Sub-Partition (ISP) (DE-LUXÁN-HERNÁNDEZ et al., 2019) divides a CU into sub-partitions, reducing the distance of the reference samples, but all partitions must use the same coding mode.

For chrominance intra coding, Cross-Component Linear Model (CCLM) (ZHANG et al., 2018) predicts the chrominance samples based on the reconstructed luminance samples by using linear models whose parameters are derived from reconstructed luminance and chrominance samples.

VVC maintains and enhances several inter-frame prediction tools of HEVC, including Advanced Motion Vector Prediction (AMVP) (LIN et al., 2013) and merge mode, and introduces novel techniques. In VVC, the Merge Candidate List (MCL) extends the prediction possibilities and includes five types of candidates in the following order: (i) Motion Vector (MV) from neighboring spatial CUs; (ii) MV from neighboring temporal CUs; (iii) history-based MV; (iv) pairwise average MV; (v) zero MV. The MCL construction process follows a specific order, and it is finished when all positions of the MCL are filled, where the maximum allowed length is six.

VVC uses the same approach of HEVC to derive spatial and temporal candidates, where up to four candidate blocks are selected from five possible spatial neighbors, and one candidate block is selected from two possible temporal neighbors (CHEN; YE; KIM, 2020). Additionally, VVC extends the prediction possibilities of the merge mode with History-based Motion Vector Prediction (HMVP) (ZHANG et al., 2019b) and pairwise average MV. In HMVP, a table stores the MVs of the last five encoded blocks maintained and updated using a first-in-first-out (FIFO) rule (BROSS et al., 2021). The pairwise average MV candidates are obtained from the average of

MVs in the first two positions of MCL. Finally, zero MVs are added in the last positions of MCL when the list is not full.

Besides the merge mode that directly derives the motion information from neighboring, historical, or zero motion information, VVC introduces Merge Mode with Motion Vector Difference (MMVD) (CHIEN et al., 2021) to refine the derived motion information transmitted by the conventional merge mode approach. In MMVD, one of the first two candidates in MCL is selected as base MV, and a Motion Vector Difference (MVD) represented by a direction and a distance is encoded as a refinement of the base MV (CHIEN et al., 2021). Four directions and two distance tables with eight entries are predefined in the encoder, which signals the selected merge candidate, direction, and distance, as detailed in (CHIEN et al., 2021).

In HEVC, motion-compensated prediction considers only the translational motion model, which cannot efficiently represent many kinds of motion present in a real-world video, such as rotation, scaling (zoom in/out), and shearing. VVC introduces the Affine Motion Compensation (AMC) (ZHANG et al., 2019a) to predict non-translational motion and improve the motion-compensated prediction efficiency. AMC describes the CU motion using MVs of two control points located at the top-right and top-left corners of the CU (four-parameter model) or three control points located at the top-right, top-left, and bottom-right corners of the CU (six-parameter model). The four-parameter model represents simpler transformations, such as rotation and scaling, and the six-parameter model is applied in more complex transformations, such as shearing. Like the conventional motion-compensated prediction, an affine AMVP and an affine merge mode are also supported in VVC for efficient prediction and coding of affine motion parameters. In the affine merge mode, VVC derives the control point MVs of the current CU based on the motion information of neighboring CUs (ZHANG et al., 2019a).

Geometric Partitioning Mode (GPM) (GAO et al., 2021a) is another new coding tool for inter-frame prediction, enabling motion compensation on non-rectangular block partitions as an alternative to the merge mode. When GPM is applied, the current CU is split into two partitions by a geometrically located straight line. In this case, each partition is predicted with motion information, and only the unidirectional prediction with the merge mode is allowed; that is, each partition has one MV and one reference frame index. In total, 64 geometric partitions are supported by the GPM encoding tool. After predicting each partition, the sample values are combined using a blending processing with adaptive weights along the geometric partition line.

Combined Inter/Intra-picture Prediction (CIIP) (CHIANG, 2018) combines an inter-frame prediction signal with an intra-frame prediction signal. When a CU is coded with the merge mode, this signal is combined with an intra-frame prediction signal generated using the Planar mode. The inter- and intra-frame prediction signals are combined using weighted averaging, where the weight value is derived based on whether above and left neighboring CUs are encoded using intra-frame prediction or not.

VVC also supports MVs that usually lead to better prediction with higher resolution than HEVC. The MVs accuracy for luminance samples supported in VVC is $^{1}/_{16}$ instead of $^{1}/_{4}$ in HEVC. Moreover, VVC adopts a scheme of adaptive resolution of MVs called Adaptive Motion Vector Resolution (AMVR) (CHIEN et al., 2021); MVs can be coded with different accuracy depending on the prediction mode. For inter-frame predicted CUs with conventional AMVP mode, the selected MV resolution can be $^{1}/_{4}$, $^{1}/_{2}$, 1, or 4, luminance samples. For CUs coded with affine AMVP mode, MV resolution can be $^{1}/_{16}$, $^{1}/_{4}$, or 1 luminance sample.

Aiming to enhance the coding efficiency of the bi-prediction, VVC introduces three techniques called Bi-prediction with CU-level Weights (BCW) (CHEN et al., 2016), Bi-directional Optical Flow (BDOF) (ALSHIN; ALSHINA, 2016), and Decoder-side MV Refinement (DMVR) (GAO et al., 2020b).

HEVC calculates the bi-prediction signal by averaging two prediction signals obtained from two different reference frames and/or using two MVs. In VVC, BCW extends this approach by applying a weighted averaging between the two bi-prediction signals, where a predefined set of weights are evaluated through the rate-distortion cost.

BDOF is a method applied in VVC to optimize the signal generated by the bi-prediction and is based on optical flow, which assumes that the movement of an object is performed smoothly. For each 4×4 sub-block in a CU, an MV is calculated to refine the motion information and minimize the displacement difference between the samples of the two reference frames for bi-prediction.

DMVR refines the bi-prediction motion of the conventional merge mode by using a bilateral search step without transmitting additional information in the bitstream. The searching process consists of an integer sample MV offset search and a fractional sample MV offset refinement process.

2.2.3. Residual and Entropy Coding

VVC enables larger transform-block sizes than HEVC, which is especially useful for higher resolution videos, such as 1080p and 4K. VVC supports transform block sizes of up to 64×64 luminance samples instead of 32×32 samples of HEVC, including square and rectangular transform block sizes that range from 4×4 to 64×64. HEVC allowed only squared transform unities with sizes ranging from 4×4 to 32×32.

Besides, VVC introduces Multiple Transform Selection (MTS) (ZHAO et al., 2016; ZHAO et al., 2021) encoding tool that enhances the transform module by including Discrete Sine Transform VII (DST-VII) and Discrete Cosine Transform VIII (DCT-VIII) to decorrelate better the prediction residual. Like HEVC, VVC applies DCT-II for horizontal and vertical directions. Still, when MTS is enabled, DST-VII and DCT-VIII can be combined, and separate transforms in horizontal and vertical directions can be applied. MTS is allowed only for 32×32 or smaller blocks and can be applied for both intra- and inter-frame predicted blocks.

In the intra-frame prediction, a secondary transform can be applied to low-frequency coefficients of blocks that selected DCT-II as primary transform to explore better the directionality characteristics of the intra-frame prediction residual signals. For this purpose, Low-Frequency Non-Separable Transform (LFNST) (KOO et al., 2019; ZHAO et al., 2021) encoding tool is applied between the forward transform and forward quantization in the encoder and between the inverse quantization and inverse transform in the decoder.

In the inter-frame prediction, Sub-Block Transform (SBT) (ZHAO et al., 2021) encoding tool allows to encode only a sub-partition of the block and skips the remaining portion. There are eight SBT partition modes available in VVC with different configurations of splits and sizes. When SBT is applied, the transform block is either half or quarter size of the residual block regarding horizontal or vertical direction, and the remaining part of the block is discarded.

The quantization module processes the residual information of the encoding block, attenuating or removing the less relevant frequencies to the human visual system; thus, causing information losses. Quantization Parameter (QP) defines the quantization level - lower QP values preserve the image details and higher QP values provide a higher compression rate at the cost of image quality losses (BUDAGAVI;

FULDSETH; BJONTEGAARD, 2014). The main novelties in this module, when compared to the HEVC, are the maximum QP value increase from 51 to 63 and Dependent Quantization (DQ), which enables the use of a second scalar quantizer (SCHWARZ et al., 2019, 2021). Furthermore, Joint Coding of Chroma Residual (JCCR) (SCHWARZ et al., 2021) allows using a single residual block for both chrominance components (Cb and Cr) when they are similar to each other.

As H.264/AVC and HEVC, the entropy coding of VVC is based on the Context Adaptive Binary Arithmetic Coding (CABAC) (MARPE; SCHWARZ; WIEGAND, 2003) that performs a well-established coding process for a significant bitstream reduction by performing lossless entropy compression at the syntax elements generated by quantization. VVC introduced some improvements, such as the multi-hypothesis probability update model, separate residual coding for transformed blocks and blocks encoded with Transform Skip Mode (TSM), and context modeling for transform coefficients (SCHWARZ et al., 2021). VVC maintains two estimators for the probability estimation and computes the average probability between them for coding. Each estimator is independently updated with different adaptation rates, which are pre-trained based on the statistics of the associated bins (SCHWARZ et al., 2021). Thus, entropy coding allows calculating the RD-cost of all possibilities of block partitioning, encoding modes, and transform combinations, enabling the encoder to select the most suitable one to predict the current block according to the encoding context.

2.2.4. In-loop Filters

VVC applies three filters in the frame reconstruction loop, which are processed in the following order: (i) Deblocking Filter (DF); (ii) Sample Adaptive Offset (SAO); and (iii) Adaptive Loop Filter (ALF). The DF filter reduces the blocking artifacts and its design is based on the one in HEVC but it is extended to consider the new block structures and coding tools with longer deblocking filters and a luminance-adaptive filtering mode designed specifically for HDR video (KARCZEWICZ et al., 2021). The SAO filter is applied to attenuate the ringing artifacts and its design is the same as in HEVC. The ALF filter was adopted in some early versions of the HEVC standardization process, but it was removed from the final standard version due to its computational complexity. Although ALF still has a high complexity, especially for the decoder, it increases coding efficiency; thus, VVC has adopted ALF after some implementation changes. ALF is applied to correct further the signal based on linear filtering and adaptive clipping (KARCZEWICZ et al., 2021). Additionally, before processing the in-loop filters, Luma Mapping with Chroma Scaling (LMCS) (LU et al., 2020) is applied to adjust the range of the luminance input sample values to improve the subjective and objective quality of the encoded video sequence.

3. VVC INTRA-FRAME PREDICTION

This chapter presents a detailed description of the VVC intra-frame prediction, which is the focus of this work, including a specific discussion of the block partitioning structure, the enhancements in the HEVC-based coding tools, and the novel coding tools for the prediction and transform processes.

3.1. Block Partitioning

As previously discussed, VVC allows split chrominance blocks separately from the luminance blocks for I-slices; a luminance CTU forms one coding tree, and a chrominance CTU (containing Cb and Cr components) forms a separate chroma tree. This coding structure is called Dual-Tree (DT) or Chroma Separate Tree (CST). This approach is motivated by the fact that luminance usually has a finer texture than chrominance, which results in a higher number of smaller CUs in luminance than those in chrominance.

The maximum and minimum luminance CU sizes processed in the intra-frame prediction are 64×64 (maximum transform block size) and 4×4 samples, respectively. Regarding chrominance, the maximum and minimum sizes are 32×32 and 16 samples (8×2 or 4×4), respectively. Besides, VVC allows chrominance blocks to have 16×2 and 32×2 samples. Blocks 4×2 and 2×*H*, where *H* is the height of the block, are excluded from addressing memory architecture and throughput requirements (HUANG et al., 2021).

Figure 4 exemplifies the QTMT partitioning applied for a 128×128 luminance CTU split into several CU sizes with different QT and MTT levels. Each colored line represents a block partition type; black lines denote the QT splitting, green and orange lines indicate the BTH and BTV splitting, respectively, and blue and red lines represent TTH and TTV splitting, respectively.

Since the intra-frame prediction is performed only for 64×64 or smaller blocks, the 128×128 luminance CTU is partitioned by QT splitting, resulting in four 64×64 CUs. Then, the QTMT depth starts from level one for the intra-frame prediction. From that, each CU can be recursively partitioned using the QT structure and each QT leaf node can be further recursively partitioned using the MTT partitioning structure.

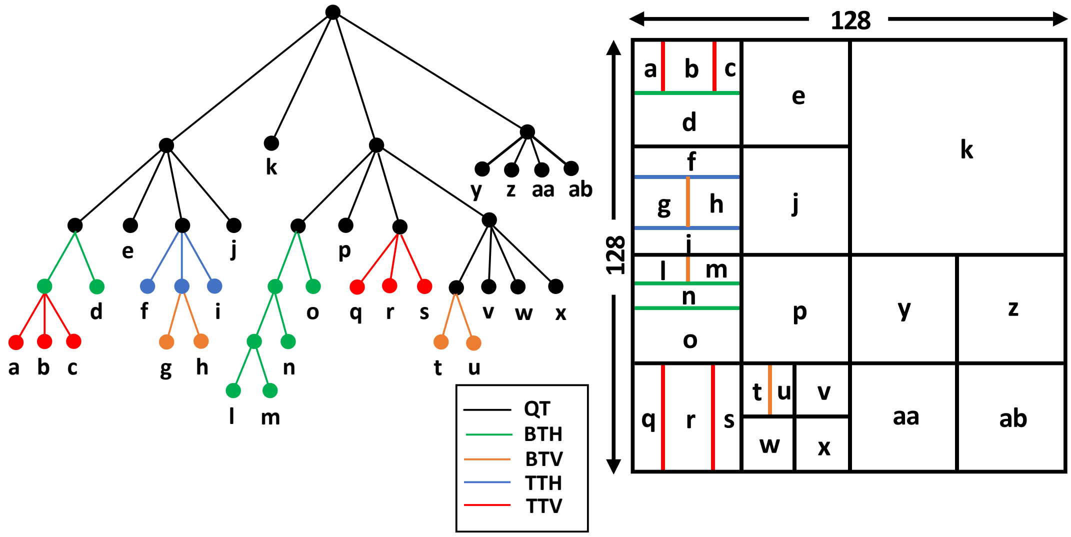


Figure 4 – Illustration of the QTMT block partitioning structure.

Figure 5 presents the CU size distribution (luminance) for the first frame of the BasketballPass video sequence encoded with VTM 10.0 (CHEN; YE; KIM, 2020; VTM, 2020) using all-intra configuration and QP 37, allowing the usage case analysis of these partition types in the intra-frame prediction. One can notice that the VVC block partition structure is strongly correlated with the image details, as expected. Figure 5(a) and (b) illustrate this effect detaching a smooth region encoded with larger block sizes and a detailed region encoded with smaller block sizes, respectively. Several MTT structure levels and different directions of BT and TT partitions are employed in the detailed region, according to the texture characteristics. In contrast, few QT and MTT splitting levels are required to provide effective compression in the smooth region.

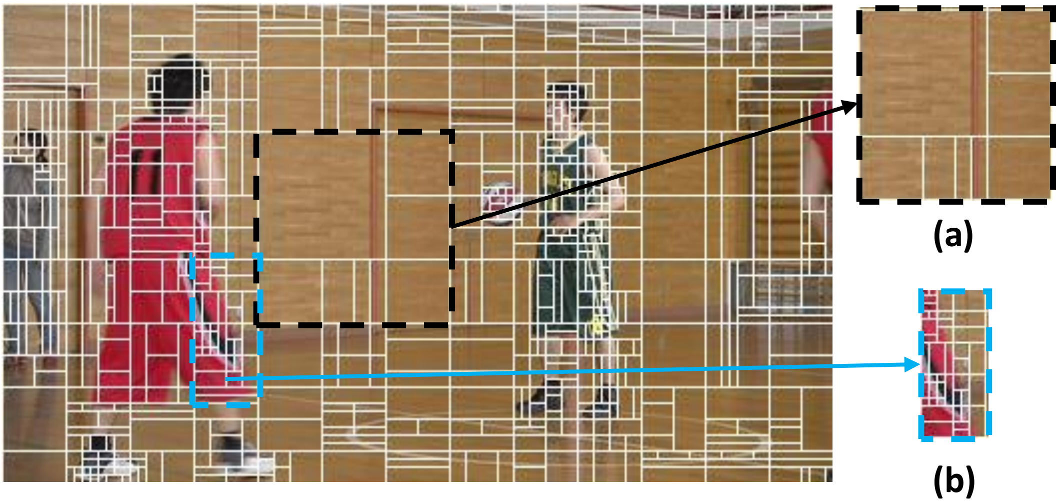


Figure 5 – CU size distribution for BasketballPass video sequence highlighting (a) smooth and (b) detailed regions.

3.2. Intra-Prediction Coding Tools

Figure 6 presents a diagram of the VVC intra-frame encoder for luminance samples; the encoder evaluates several encoding modes to minimize the RD-cost. Thus, the encoding process selects the prediction mode that reaches the lowest RD-cost. Similar to the HEVC Test Model (HM) (ROSEWARNE et al., 2015; HM, 2018), the intra-frame prediction of the VTM employs Rough Mode Decision (RMD) and Most Probable Modes (MPM) (ZHAO et al., 2011) to build a list of promising candidates named Rate-Distortion list (RD-list). RMD performs a local evaluation to estimate the encoding cost of each candidate mode instead of evaluating all encoding possibilities by their RD-cost using the RDO, which involves more complex operations, implying a prohibitive computational effort.

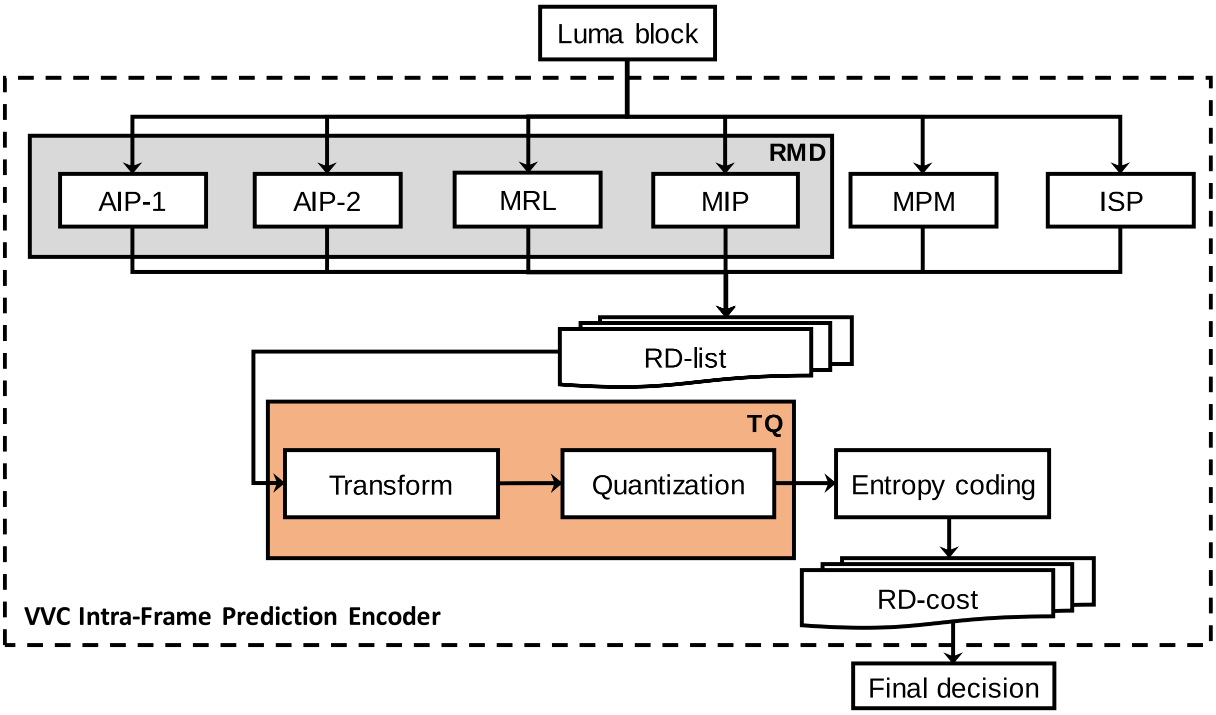


Figure 6 – Diagram of the VVC intra-frame prediction for luminance blocks.

RMD estimates the required bits to encode the prediction mode and the encoding cost through the Sum of Absolute Transformed Differences (SATD) (between the original and predicted block samples). Then, the algorithm orders the modes according to their SATD-based costs and inserts a few modes with the lowest costs ordered into the RD-list. Following, MPM gets the default modes (the most frequently used ones), and the modes in the left and above neighbor blocks and inserts at most two additional modes into the RD-list. For all-intra encoder configuration, the RD-list starts with sizes of 8, 7, and 6 modes for 64×64, 32×32 and the remaining blocks

(32×16, 16×32, 16×16, 32×8, 8×32, 32×4, 4×32, 16×8, 8×16, 8×8, 16×4, 4×16, 8×4, 4×8, and 4×4), respectively. However, the final size of the RD-list can vary significantly according to the block size since it changes dynamically based on the encoding context and the use of fast decisions.

Angular Intra Prediction-1 (AIP-1) in Figure 6 represents the same HEVC intra-frame prediction modes, but VVC also brings novel intra-frame coding modes compared to HEVC, including Angular Intra Prediction-2 (AIP-2) (BROSS et al., 2021), Multiple Reference Line (MRL) (CHANG et al. 2019), Matrix-based Intra Prediction (MIP) (SCHAFER et al., 2019), Intra Sub-Partition (ISP) (DE-LUXÁN-HERNÁNDEZ, 2019), and an intra mode coding method with six MPMs instead of three MPMs as in HEVC (PFAFF et al., 2021). In the VVC intra coding method with six MPMs, Planar mode is always coded as the first MPM, DC and angular modes are coded using the remaining five positions of the MPMs list that are derived from intra prediction modes from left and above neighboring blocks.

After processing these prediction tools, the modes inserted into the RD-list are processed by the residual coding, including Transform and Quantization (TQ) steps. Subsequently, the RD-costs are obtained applying Entropy Coding (EC) in the TQ flow results. The transform module encompasses Multiple Transform Selection (MTS) (ZHAO et al., 2016) and Low-Frequency Non-Separable Transform (LFNST) (KOO et al., 2019) as will be detailed in Section 3.2.6.

For chrominance blocks, VVC inherits the HEVC prediction modes and inserts Cross-Component Linear Model Prediction (CCLM) (ZHANG et al., 2018), where chrominance samples are predicted based on the reconstructed luminance samples by using a linear model.

The next sections present the novel intra-prediction tools defined in VVC.

3.2.1. Angular Intra Prediction

The HEVC angular prediction modes are extended from 33 to 65 angular modes to represent various texture patterns and provide higher accuracy for intra-frame prediction. Figure 7 illustrates the VVC angular intra-frame prediction modes, where the solid black lines depict the modes already used in HEVC intra-frame prediction, and dotted red lines are the ones introduced in VVC. Adding Planar and DC modes, the number of intra-frame prediction modes has increased to 67. These two modes remain with the same approach used in HEVC. Although Planar and DC are non-angular prediction modes, we call this tool of Angular Intra Prediction (AIP) for simplicity.

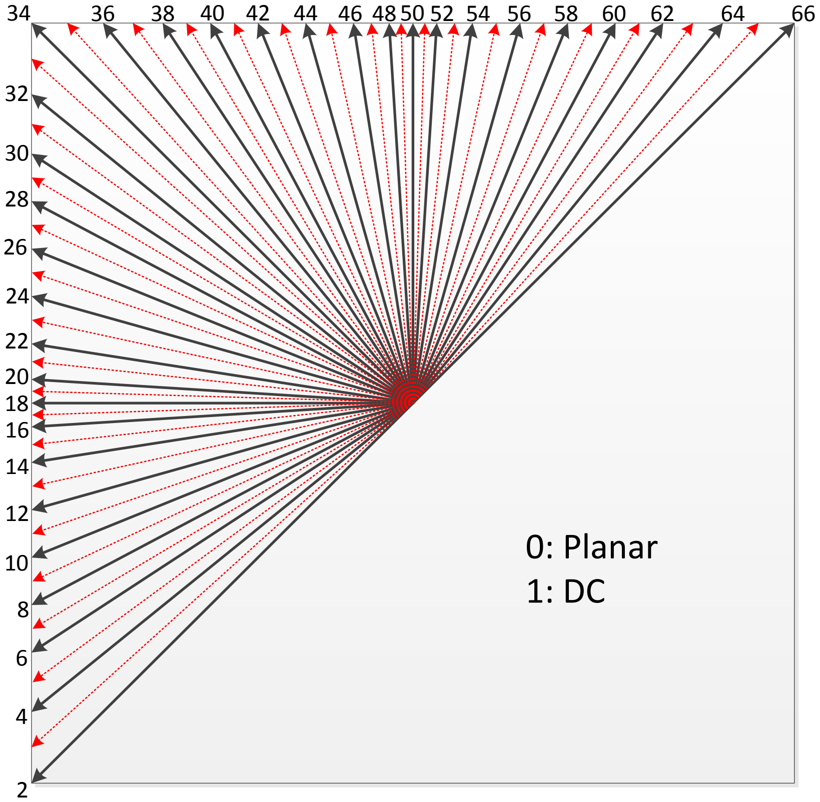


Figure 7 – VVC angular intra-frame prediction modes (CHEN; YE; KIM, 2020).

The VTM encoder divides AIP into AIP-1 and AIP-2 steps to avoid an exhaustive evaluation of the 67 intra-frame prediction modes for each available block size. AIP-1 evaluates through the RMD process the Planar, DC, and 33 angular modes inherited from HEVC (solid black lines in Figure 7) and inserts a few modes into the RD-list. AIP-2 evaluates the angular modes adjacent to the angular modes already included in the RD-list (i.e., the best modes selected in AIP-1) and orders the RD-list based on the obtained SATD-based costs of these two steps. Thus, a reduced set of the new VVC angular intra-frame prediction modes is evaluated (CHEN; YE; KIM, 2020).

Since VVC allows the intra-frame prediction of rectangular blocks due to the CU shapes obtained in the QTMT partitioning structure, it includes Wide-Angular Intra Prediction (WAIP) (ZHAO et al., 2019) coding tool to handle these block shapes properly. This tool was developed because the 35-conventional angular modes were defined targeting squared blocks, and good prediction samples may not be reached for rectangular blocks because of the predefined angles. Thus, if the block width is larger than the block height, prediction modes with angles beyond 45 degrees in the top-right direction are evaluated. Otherwise, if the block height is larger than the block width, prediction modes with angles beyond 45 degrees in the bottom-left direction are evaluated. WAIP does not increase the number of intra-frame prediction modes evaluated since, in this case, these wide-angle modes replace the prediction modes in the opposite direction with conventional angles (CHEN; YE; KIM, 2020). Additionally, for rectangular blocks, the DC prediction mode considers only the larger block side samples to provide a computationally efficient implementation.

3.2.2. Multiple Reference Line Prediction

The MRL prediction (CHIANG et al., 2019) allows more reference lines for the VVC intra-frame prediction than the ones used in HEVC. Figure 8 shows a block size of 4×4 samples and the reference lines used in the VVC intra-frame prediction when the MRL prediction is enabled.

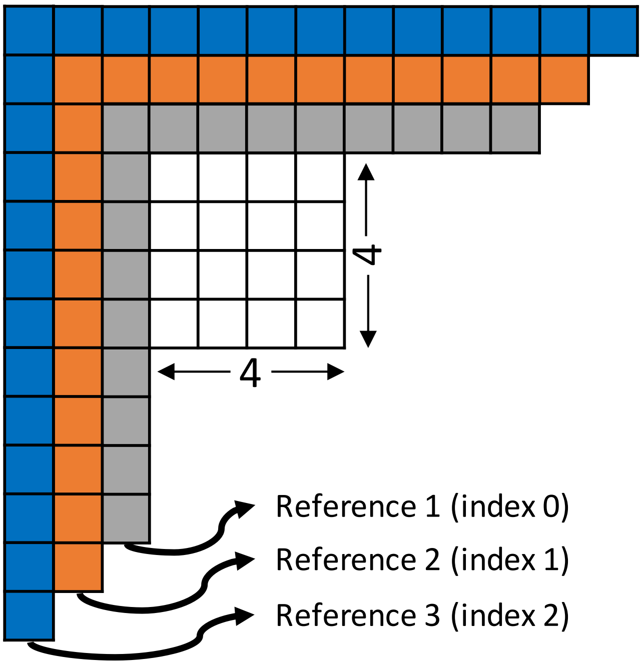


Figure 8 – Illustration of the MRL intra-frame prediction.

Reference 1 (index 0) refers to the nearest reference line, which is considered for the AIP tool. References 2 and 3 (indexes 1 and 2) are the two additional reference lines evaluated by the MRL tool. The evaluation of these two farther reference lines can improve the coding efficiency of the intra-frame prediction since the adjacent reference line may significantly differ from the predicting block due to discontinuities, leading to a meaningful prediction error.

The MRL tool evaluates each combination of prediction mode and reference line using RMD and updates the RD-list (which already contains the best modes selected in the AIP tool). However, evaluating all available intra-frame prediction modes with this extra number of reference lines increases the encoder complexity significantly. Thus, the MRL prediction evaluates only MPMs, excluding Planar mode, for the two extra reference lines (reference lines 2 and 3) (PFAFF et al., 2021). Planar mode is not considered for the MRL encoding tool since this combination does not provide additional coding gain for the VVC intra-frame prediction (PFAFF et al., 2021).

3.2.3. Matrix-based Intra Prediction

MIP (SCHAFER et al., 2019) is an alternative approach to the conventional angular intra-frame prediction modes, representing a new concept of intra predictors designed by data-driven methods (PFAFF et al., 2021). MIP performs the intra-frame prediction through matrix multiplication and samples interpolation. Figure 9 demonstrates the MIP process for a block of 8×8 samples, where neighboring samples of the adjacent reference lines are also used as prediction input. These neighboring samples are subsampled to perform the matrix multiplication, followed by the addition of an offset (b_k) and a linear interpolation (horizontal and vertical) to obtain the predicted block (SCHAFER et al., 2019). A set of matrices were defined according to the block size by offline training through neural networks, and each matrix represents a prediction mode.

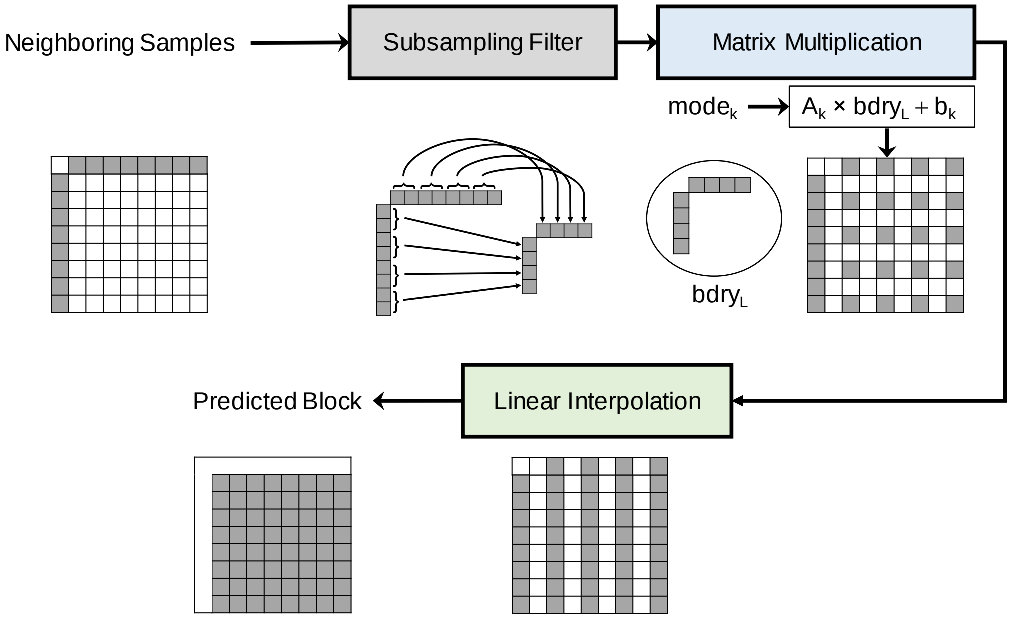


Figure 9 – Matrix-based intra prediction flow for a block size of 8×8 samples.

MIP allows 16 matrices for 4×4 blocks, eight matrices for 8×8 blocks and blocks with exactly one side of length 4, and six matrices for the remaining block sizes. Besides, the transposed matrices are also considered, doubling the number of prediction modes for each set. These prediction modes are also evaluated using RMD, and the RD-list is updated with the lowest SATD-based costs among AIP, MRL, and MIP prediction modes (CHEN; YE; KIM, 2020).

MIP improves the encoding efficiency enabling predictions that vary in more than one direction (i.e., non-linear prediction), which is impossible with conventional angular modes.

3.2.4. Intra Sub-Partition

ISP (DE-LUXÁN-HERNÁNDEZ et al., 2019) explores the correlations among block samples to improve the VVC intra-frame prediction. ISP divides the current block horizontally or vertically into sub-partitions sequentially encoded using the same intra-frame prediction mode. The sub-partitions are processed from top to bottom (horizontal split) or left to right (vertical split). The reconstructed samples of each encoded sub-partition are used as reference samples for the next sub-partition, increasing the reference sample correlation compared to the conventional approach, which locates the reference samples at the left and above boundaries of the predicting block. Figure 10 exemplifies ISP for a 16×16 block split into 16×4 horizontal and 4×16 vertical sub-partitions.

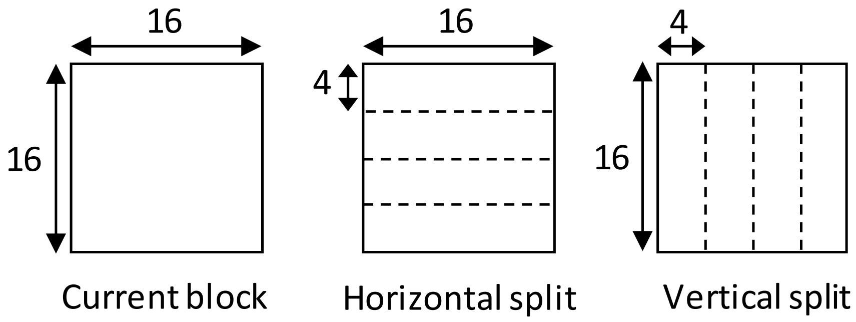


Figure 10 – Intra Sub-Partition for a 16×16 block split into 16×4 horizontal and 4×16 vertical sub-partitions.

VVC keeps the 16-samples throughput splitting 4×8 and 8×4 blocks into two sub-partitions instead of four and disabling ISP for 4×4 blocks; ISP splits the current block with the remaining sizes into four sub-partitions.

ISP cannot use the RMD process since it requires the real reconstructed samples used as a reference to get the next sub-partition prediction, which can only be obtained by performing the complex RDO process. Consequently, VTM adopts some strategies to derive the most promising prediction modes. Firstly, RDO is performed with the RD-list containing the best SATD-based costs among AIP, MRL, and MIP. Thus, ISP can use the SATD-based costs and RD-costs of the AIP tool (including AIP-1, AIP-2, and WAIP) to build a promising candidate list. The MRL and MIP tools are not considered for ISP mode derivation; then, the ISP list is generated alternating the split types (horizontal and vertical) in the following order: (i) Planar, (ii) angular modes ordered by RD-cost, (iii) DC, and (iv) the best AIP SATD-based costs discarded of the RD-list after processing MRL and MIP. Thus, the ISP list can derive up to 16 prediction modes being the same eight modes for the horizontal and vertical split.

3.2.5. Cross-Component Linear Model

The VVC intra-frame prediction for chrominance blocks considers a list of eight candidate modes: (i) Planar (mode 0); (ii) horizontal (mode 18); (iii) vertical (mode 50); (iv) DC (mode 1); (v) CCLM_LT; (vi) CCLM_L; (vii) CCLM_T; and (viii) Derived Mode (DM). The first four prediction modes are the same applied to luminance samples, but in this case, only the chrominance samples are considered for prediction. The last four prediction modes explore the correlation between the luminance and chrominance components, where chrominance samples are predicted using information present in the luminance prediction.

The CCLM prediction modes use the reconstructed luminance samples as references to predict the chrominance samples, using a linear model as follows

$$P(i,j) = a \cdot rec'_L(i,j) + b \tag{1}$$

where $P(i,j)$ refers to the predicted chrominance samples and $rec'_L(i,j)$ represents the reconstructed luminance samples. The parameters a and b are derived based on reconstructed neighboring luminance and chrominance samples (ZHANG et al., 2018).

Three CCLM prediction modes are supported in VVC, including CCLM_LT, CCLM_L, and CCLM_T. These prediction modes differ only in the locations of the reference samples that are used for model parameter derivation, where CCLM_L uses reference samples from the left boundary, CCLM_T involves reference samples from the top boundary, and CCLM_LT uses reference samples from both the top and left boundaries.

DM refers to the prediction mode derived from the collocated luminance block; i.e., a prediction mode that has already been selected to encode the luminance block is inserted as a candidate to perform the prediction of the chrominance block. If the derived mode is equal to any conventional prediction modes applied to luminance blocks (first four modes: planar, horizontal, vertical, or DC), mode 66 (diagonal 45º, see Figure 7) is added in place of this conventional mode. If the collocated luminance block uses MIP, the Planar mode is applied, except for the 4:4:4 chrominance color format with a single partitioning tree that the same MIP mode is applied for the chrominance block (PFAFF et al., 2021).

3.2.6. Transform Coding

VVC enhances the transform coding for the intra-frame prediction by including Multiple Transform Selection (MTS) (ZHAO et al., 2016) and Low-Frequency Non-Separable Transform (LFNST) (KOO et al., 2019), which are tools for primary and secondary transform modules, respectively. Figure 11 presents the transform coding process for the VVC intra-frame prediction of luminance samples. The RD-list defined by the intra-prediction steps are the inputs for the transforms, and VVC enables combining different transforms intending to minimize the RD-cost.

In addition to DCT-II, used as the main transform in HEVC, VVC also enables using DST-VII and DCT-VIII, increasing the contribution of this module to the global coding efficiency, but raising a lot the computational effort required to process the transforms, if compared to HEVC. VVC also defines the use of Transform Skip Mode (TSM), as in HEVC. TSM is available for 32×32 or smaller blocks; in this case, the prediction residues are directly sent to the quantization step, avoiding the use of transforms.

VVC transform coding has three main processing paths regarding the primary transform application, as presented in Figure 11: (i) the first one using DCT-II for horizontal and vertical directions; (ii) the second one using TSM, and (ii) the third one using MTS, where DST-VII and DCT-VIII are used. The paths (i) and (ii) are similar to the HEVC transforms. The use of MTS in the path (iii) allows a combination of DST-VII and DCT-VIII in horizontal and vertical directions; then, four combinations are evaluated: (i) DST-VII and DST-VII, (ii) DST-VII and DCT-VIII, (iii) DCT-VIII and DST-VII, and (iv) DCT-VIII and DCT-VIII (ZHAO et al., 2021). The VTM implementation process these steps sequentially, starting with DCT-II, then TSM, and later the MTS processing.

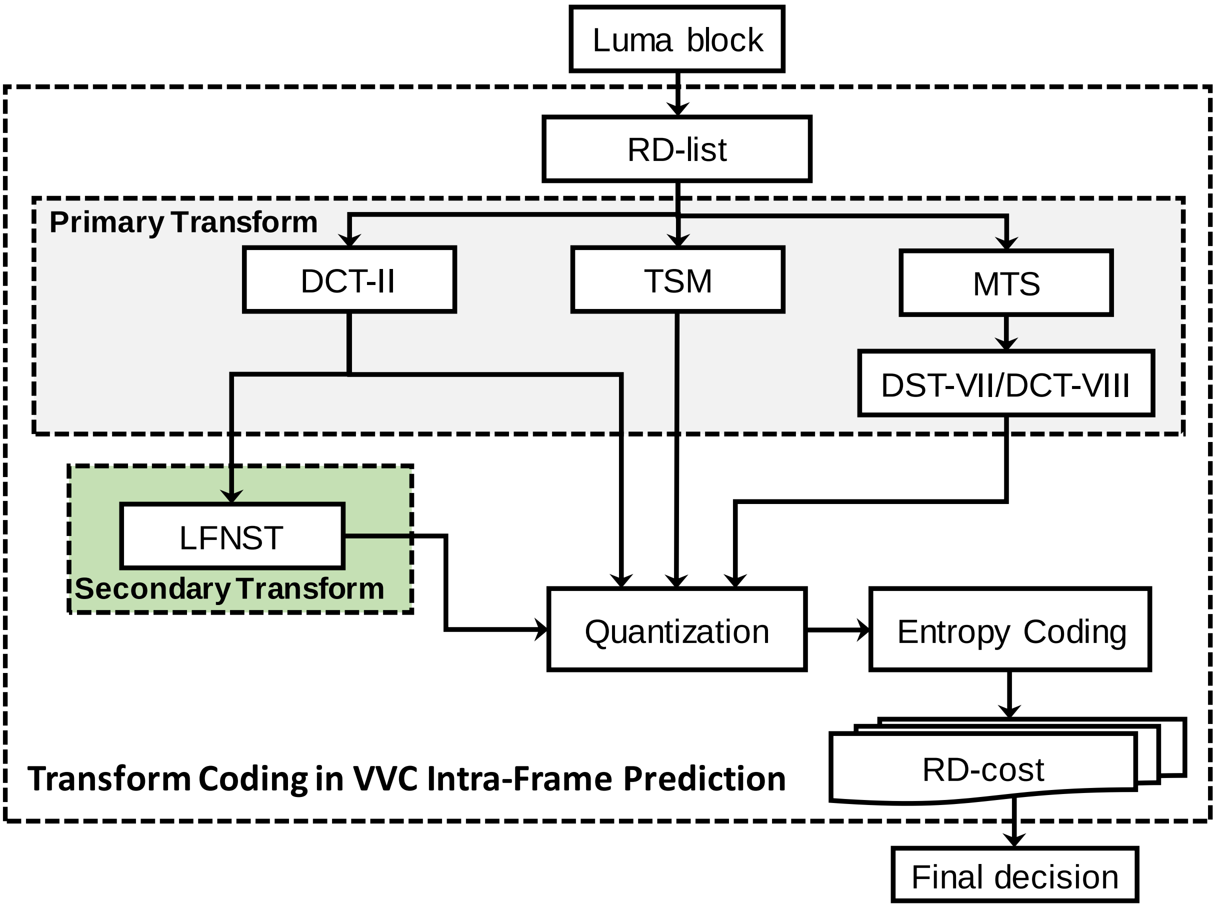


Figure 11 – Diagram of transform coding in VVC intra-frame prediction.

The primary transform block sizes are different where DCT-II has sizes ranging from 4×4 to 64×64, and DST-VII and DCT-VIII have sizes ranging from 4×4 to 32×32. All these cases enable square and rectangular shapes, and MTS is applied only for luminance samples.

The ISP coding tool is an exception for the processing flow of the primary transforms; it enables combinations between DCT-II and DST-VII and decides which transform will be used in horizontal or vertical direction considering the width and height of the processed sub-partition. For other prediction tools, combinations of DCT-II and DST-VII/DCT-VIII are not allowed due to the limited coding gain and increased complexity for introducing additional encoding evaluations with more transform combinations (ZHAO et al., 2021).

High-frequency coefficients are zeroed out for transforming blocks with sizes (width or height, or both width and height) equal to 64 for DCT-II and 32 for DCT-VIII and DST-VII to decrease the computational complexity. Thus, only low-frequency coefficients are retained.

LFNST is a non-separable secondary transform of the VVC intra-frame prediction that further decorrelates the low-frequency primary transforms coefficients (top-left region of the transform block). LFNST may be applied for luminance blocks ranging from 4×4 to 64×64 samples, including square and rectangular block shapes. However, LFNST can be applied only for intra coded blocks that use DCT-II as primary transform, as presented in Figure 11. LFNST contains two secondary transform sets (LFNST 1 and LFNST 2) with four non-separable transform matrices for each set (ZHAO et al., 2021). The transform matrix evaluated for each set is defined based on the intra-frame prediction mode (ZHAO et al., 2021). The VTM process the evaluation of DCT-II without LFNST (also referred to as LFNST 0), DCT-II with LFNST 1, followed by DCT-II with LFNST 2. When LFNST is not applied, the DCT-II results are sent directly to quantization. Analogously to the luminance blocks, LFNST also can be applied for chrominance blocks.

4. RELATED WORKS

This chapter presents a collection of works found in the literature that propose solutions for reducing the computational effort of the VVC encoder based on heuristic and machine learning techniques. These solutions were developed targeting different steps of the VVC intra-frame prediction, including block partitioning, intra mode decision and transform coding. All related works mentioned in this chapter were evaluated under all-intra encoder configuration.

Fu et al. (2019a) propose a fast block partitioning algorithm using a classifier based on the Bayesian decision rule. The information derived from the current block and horizontal binary splitting is used as an input feature for the classifier, responsible for deciding when skipping the vertical split types. Additionally, the horizontal ternary split is skipped if the cost of the vertical binary split is lower than the cost of the horizontal binary split. Their solution was implemented in VTM 1.0, reducing 45% the encoding time and increasing 1.02% the Bjontegaard Delta Bit Rate (BDBR) (BJONTEGAARD, 2001).

The work of Yang et al. (2020) presents a scheme composed of a fast block partitioning solution based on decision tree classifiers and a fast intra mode decision to reduce the number of angular intra-frame prediction modes evaluated. They trained one decision tree classifier for each split type using the texture information of the current and neighboring blocks. Since the classifiers are responsible for deciding the best split type before predicting the current block size (i.e., not split type), they used only texture information features of the current and neighboring blocks. The proposed scheme was implemented in VTM 2.0. The fast block partitioning solution saves 52.59% of encoding time for a 1.56% of BDBR increase. The fast intra mode decision reduces 25.51% of the encoding time and increases 0.54% of the BDBR. The overall solution provides 62.46% of encoding time saving with a 1.93% of BDBR increase.

Chen et al. (2020a) present a solution to reduce the VVC intra coding time using SVM classifiers to decide between horizontal and vertical partitioning. Six classifiers are trained online using only texture information of the current block during the first frame encoding; the remaining frames are encoded, applying the decisions of the trained classifiers. The proposed solution was implemented in VTM 2.1, providing 50.97% encoding timesaving with a BDBR increase of 1.55%.

Lei et al. (2019) present a fast solution to decide in advance the direction of

BT and TT partitions. Their solution evaluates a subset of directional intra-frame prediction modes for virtual sub-partitions of the current block to estimate the horizontal and vertical splitting costs of the current block. Based on the estimated costs, their solution can decide by skipping horizontal or vertical partitions. Their approach, implemented in VTM 3.0 and, saves 45.8% of the encoding time with a 1.03% BDBR increase.

The work of Zhang et al. (2020) proposes a scheme composed of two solutions, the first one avoids the evaluation of some block partitions and the second one reduces the number of AIP modes evaluated in the intra-frame prediction. For the block partitioning, the proposed solution was defined in two steps, where the first step classifies the texture CU as homogeneous or complex. The texture complexity is measured based on the difference and standard deviation of block samples, and the classification is performed through a comparison with predefined threshold values. When a CU is homogeneous, the splitting process is finished; otherwise, Random Forest (RF) classifiers are applied to define the split type to evaluate, including no split, QT, BTH, BTV, TTH, and TTV. To reduce the number of AIP modes evaluated in the RMD process, the second solution divides these modes into four sets based on their directions. After, the Canny edge detector algorithm is applied to identify the texture direction and to define only two sets of prediction modes to be evaluated. Finally, the SATD costs of MPM modes are computed and the prediction modes with SATD cost higher than the MPMs are discarded, except Planar, DC, horizontal, and vertical modes. The proposed scheme was implemented in VTM 4.0, reaching 54.91% encoding time reduction with a 0.93% BDBR increase.

The work of Tang et al. (2019) presents solutions to reduce the computational effort of both intra- and inter-frame predictions, but in this section, only the solution for intra-frame prediction is discussed. The proposed solution for intra-frame prediction applies Canny edge detector algorithm to identify the texture direction of the block and to classify the block texture complexity based on predefined threshold values. When the block is classified as homogenous, the partitioning process is finished; otherwise, the texture orientation is verified and classified as horizontal or vertical. When the algorithm classifies the texture as vertical, only BTV and TTV split types are evaluated. In contrast, only BTH and TTH split types are evaluated when the texture is classified as horizontal. This solution was implemented in VTM 4.0.1rc1, providing 36.18% of encoding timesaving at the cost of 0.71% BDBR increase.

Cui et al. (2020) present a scheme to reduce the computational effort of VVC intra coding based on the direction of the sample gradients to decide the best block partitioning structure. Their scheme performs the decision on three partitioning possibilities, including split or not, horizontal or vertical, and BT or TT. To perform the decision, the gradients of current block sub-partitions are computed in four directions and compared with predefined threshold values. Their scheme was implemented in VTM 5.0, reaching 50.01% of encoding timesaving with an increase of 1.23% in the BDBR.

The works of Tissier et al. (2020), Zhao et al. (2020), and Li et al. (2021a) propose encoding timesaving solutions based on the Convolutional Neural Network (CNN) to define the best block partitioning. The solution of Tissier et al. (2020) was implemented in VTM 6.1. The experimental results showed 42.2% encoding timesaving with a 0.75% BD-rate increase. The solutions presented in (ZHAO et al., 2020; LI et al., 2021a) were implemented in VTM 7.0. The first solution reduces the encoding time by about 39.39%, with a BD-rate increase of 0.86%; the second one saves 46.13% of encoding time with a 1.32% BD-rate increase.

The work of Fan et al. (2020) proposes a solution based on the current block variance, sub-partition variances, and Sobel Filter. The current block variance is computed to check the homogeneity of 32×32 blocks and early terminate the QTMT evaluation. The sub-partition variances are calculated to choose only one split among QT, BTH, BTV, TTH, and TTV. The Sobel filter is used to decide by skipping the BT/TT partitions and evaluating only the QT partitioning. Their solution was implemented in VTM 7.0. The proposed solution reached a 49.27% encoding timesaving with a 1.63% BDBR increase.

Li et al. (2021b) developed a timesaving solution to skip binary and ternary splitting based on residual block variances of sub-partitions, obtained through the absolute difference between original and predicted samples. The absolute difference between variances of vertical and horizontal sub-partitions is computed and compared with predefined threshold values to early skip BT and TT evaluations. Their solution was implemented in VTM 7.1, saving 43.9% of the encoding time with a 1.50% BDBR increase.

The work of Zhang et al. (2019c) presents a technique to reduce the computational cost of arithmetic operations in the transform coding. For this purpose, a fast algorithm for DST-VII and DCT-VIII was developed. This algorithm explores the correlation of the transform coefficients to reduce the number of operations, such as the reuse of calculations with coefficients that are only with signal changes or flipped positions. The proposed solution was implemented in VTM 1.1 and evaluated with all-intra, random-access, and low delay B configurations. The results for all-intra configuration demonstrated that this solution could reduce the encoding time by 3% without impacting the coding efficiency.

Fu et al. (2019b) propose an early decision scheme for the MTS encoding tool focusing on the intra-frame prediction. This scheme verifies if all the neighboring spatial blocks were encoded with DCT-II to skip the DST-VII and DCT-VII evaluations; otherwise, the frequency of the transforms used in the neighboring blocks is computed and the transforms are evaluated from the most frequent to the least frequent. In this case, if an intra-frame prediction mode in the current transform evaluation obtained a higher RD-cost than in the previously evaluated transform, that prediction mode is discarded from the next transform evaluations. This solution was implemented in VTM 3.0, saving 23% encoding time with a 0.16% BDBR increase.

4.1. Summary of Related Works

Table 1 summarizes the 14 related works focusing on reducing the computational effort of the VVC encoder. This table indicates the module of the VVC encoder considered in each solution, including QT, MTT, Mode, and Transform (T). The column “Mode” indicates that the work reduces the computational effort of selecting intra prediction modes, such as reducing the number of AIP modes evaluated in the RMD process. Furthermore, this table provides the VTM version employed in the experiments and BDBR and Encoding Time Saving (ETS) results. The works listed in this table are sorted by VTM version.

One can notice from this table that most of these works focused on the partitioning structures (MTT and QT) for intra-frame prediction. Although these works have been recently published, these solutions targeted old VTM versions that did not include several standardized coding tools. Besides, most of these works focus only on a specific module of the encoder. Therefore, since VVC is a recent video coding standard and few works have been developed so far, there is a vast space for further research and proposals of new solutions to reduce the VVC encoder computational effort. The fast encoding solutions developed in this Thesis aim to fill this gap by proposing novel solutions exploring heuristics and machine learning techniques for block partitioning, intra-frame prediction mode selection, and transform mode selection able to achieve impressive encoding time reduction with minimal impact on coding efficiency, considering all novel VVC intra coding tools.

Table 1 – Summary of the related works focusing on reducing the computational effort of the VVC intra-frame prediction.

Work	QT	MTT	Mode	T	VTM	BDBR (%)	ETS (%)
(FU, 2019a)		×			1.0	1.02	45.00
(ZHANG, 2019c)				×	1.1	0.00	3.00
(YANG, 2020)	×	×	×		2.0	1.93	62.46
(CHEN, 2020a)		×			2.1	1.55	50.97
(LEI, 2019)		×			3.0	1.03	45.80
(FU, 2019b)				×	3.0	0.16	23.00
(ZHANG, 2020)	×	×	×		4.0	0.93	54.91
(TANG, 2019)	×	×			4.0.1	0.71	36.18
(CUI, 2020)	×	×			5.0	1.23	51.01
(TISSIER, 2020)	×	×			6.1	0.75	42.20
(ZHAO, 2020)	×	×			7.0	0.86	39.39
(LI, 2021a)	×	×			7.0	1.32	46.13
(FAN, 2020)	×	×			7.0	1.63	49.27
(LI, 2021b)		×			7.1	1.50	43.90

5. PERFORMANCE ANALYSIS OF VVC INTRA-FRAME PREDICTION

This chapter presents the VVC intra-frame prediction performance analysis. Section 5.1 introduces the methodology used in the experiments, Section 5.2 presents a computational effort and compression performance evaluation between VVC and HEVC, Section 5.3 displays the encoding time distribution of luminance and chrominance components, Section 5.4 analyzes the block size time and usage distribution, Section 5.5 shows encoding-mode time and usage distribution, Section 5.6 shows transform time and usage distribution, and Section 5.7 presents a rate-distortion and computational effort evaluation of VVC intra-frame coding tools. Finally, Section 5.8 presents a general discussion, indicating possibilities for reducing the computational effort of VVC intra-frame prediction. The results of these analyses were published in (SALDANHA et al., 2021a).

5.1. Methodology

All the analyses presented in this chapter followed the Common Test Conditions (CTC) (BOSSEN et al., 2020) for Standard Dynamic Range (SDR) video sequences with chrominance subsampling configuration 4:2:0 specified by JVET. The experiments considered the all-intra configuration, where only intra-frame prediction tools are available. The experiments were executed using the VTM software (version 10.0), which serves as a reference implementation of all the encoding features defined in the VVC standard. VTM implements all encoding tools defined in VVC and implements some heuristics to reduce the encoder complexity, as discussed later. It is important to highlight that although new VTM versions are still being released with some new functionalities for different applications (e.g., high bit depth and other chrominance formats), the intra coding flow for conventional camera-captured video sequences in VTM 10.0 remains without modifications and provides the same performance results compared to the latest VTM version. The experiments for HEVC were executed in HM software (version 16.20) with evaluations regarding the all-intra encoder configuration. For both VTM and HM encoder configurations, the default temporal subsampling factor of 8 was considered (i.e., the encoding process is performed at every 8 frames).

The CTC specification has been developed to be a benchmark to evaluate coding tools and to allow a fair comparison of different techniques. The video sequences specified to be encoded inside CTC contain several distinct characteristics to provide a robust evaluation. The CTC specifies the encoding of six classes of video sequences with distinct video resolutions. Classes A1 and A2 refer to six UHD 4K (3840×2160 resolution) video sequences. Class B has five video sequences with 1920×1080 resolutions. Class C and D represent videos with 832×480 and 416×240 resolutions, respectively, each one with four video sequences. Finally, Class E indicates three video sequences with 1280×720 resolution, totalizing 22 video sequences. Moreover, each video sequence should be encoded with 22, 27, 32, and 37 QP values. More details about the CTC used in the experiments for VVC are provided in APPENDIX A of this Thesis.

The compression performance and computational effort were measured using BDBR and encoding time, respectively. Modifications and additional functions were included in the VTM reference software for obtaining the encoding time and usage distributions of the encoding tools.

5.2. VVC vs HEVC Compression Performance and Computational Effort

The first experiment compares the compression performance and the encoding time of VTM and HM. Figure 12 presents the compression efficiency of VTM for luminance (Y) and chrominance components (Cb and Cr) compared to HM, considering all-intra encoder configuration.

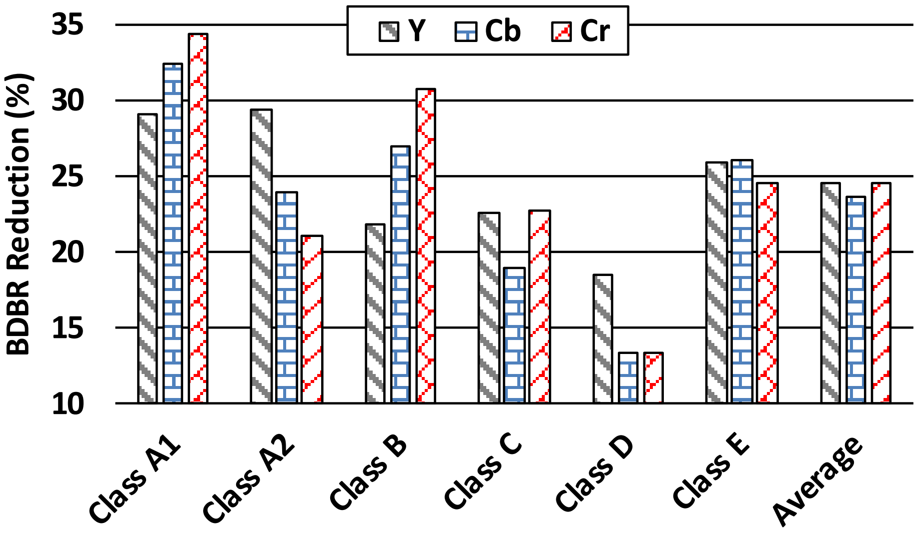


Figure 12 – Compression Efficiency of VTM compared to HM for the all-intra configuration.

One can notice from Figure 12 that VTM performs better than HM for all cases regarding video resolution and encoding components. For luminance and chrominance components VTM obtains a BDBR reduction of up to 29.3% (Class A2) and 34.4% (Class A1), respectively. Considering the luminance component, the smallest BDBR reduction is noticed for Class D (18.5%), whereas the highest BDBR reductions are observed for Classes A1 and A2. This noticeable gain for high-resolution video occurs mainly due to the larger block sizes and block partitioning structure, which allows the encoding of larger blocks for uniform regions and more flexible partition types for detailed regions. Thus, this experiment showed that VVC can provide an important higher compression rate than HEVC for intra-frame prediction, especially for high-resolution videos.

Figure 13 displays the encoding time increase for each class of video sequences and QP values.

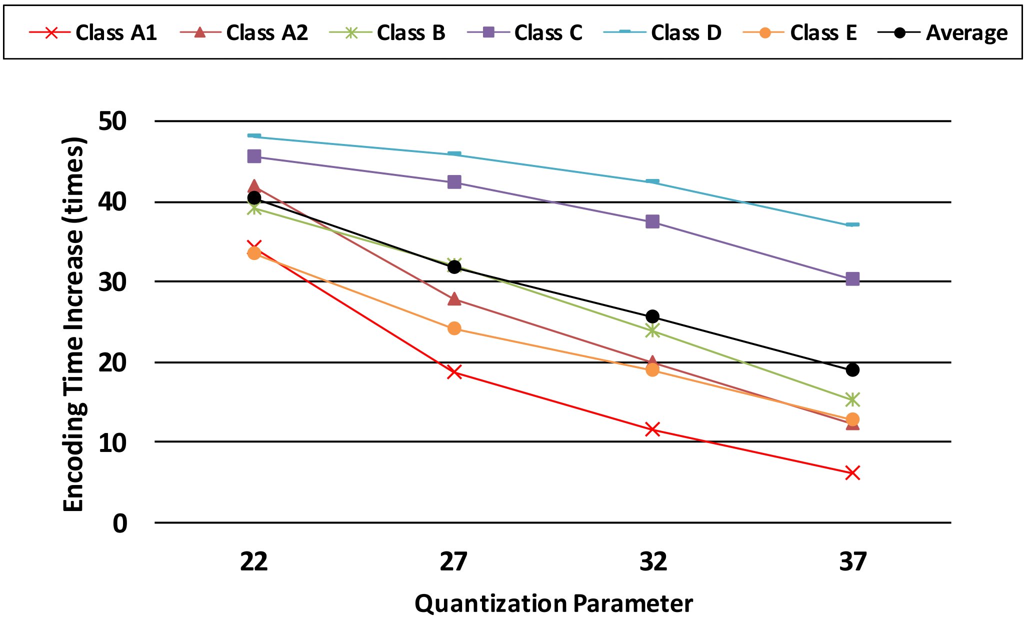


Figure 13 – Encoding time increase of VTM compared to the HM for the all-intra configuration.

One can notice that QP=22 has the highest encoding time increase rates for all classes, where VTM is 40 times slower than HM, on average. This result is expected because VTM performs several evaluations of block sizes and intra-frame prediction modes to preserve more image details for lower QPs. The encoding time of VTM compared to HM increases about 32, 26, and 19 times for QPs equal to 27, 32, and 37, respectively. Classes C and D showed the highest encoding time increase over HM (39 and 43 times, on average) because lower video resolutions tend to be encoded with smaller block sizes, implying the QTMT expansion to evaluate several combinations of block sizes and prediction modes. In contrast, higher video resolutions tend to be encoded with larger block sizes, and fast decisions can avoid expanding QTMT early. On average, VTM takes 27 times more encoding time than HM for intra-frame prediction.

This analysis proved that the enhancements performed in VVC intra-frame prediction enable a higher compression rate, outperforming the compression efficiency of HEVC considerably and enabling the transmission of high-resolution video at a lower bit rate. However, this efficiency requires a very high computational effort, hampering the real-time video coding.

5.3. Computational Effort Distribution of Luminance and Chrominance

Figure 14 displays the VTM intra-frame encoding time distribution, considering luminance and chrominance components and the QP values defined in CTC. Since the encoder enables the use of distinct coding tree structures for both components, this analysis allows identifying the impact of each channel in the total encoding time.

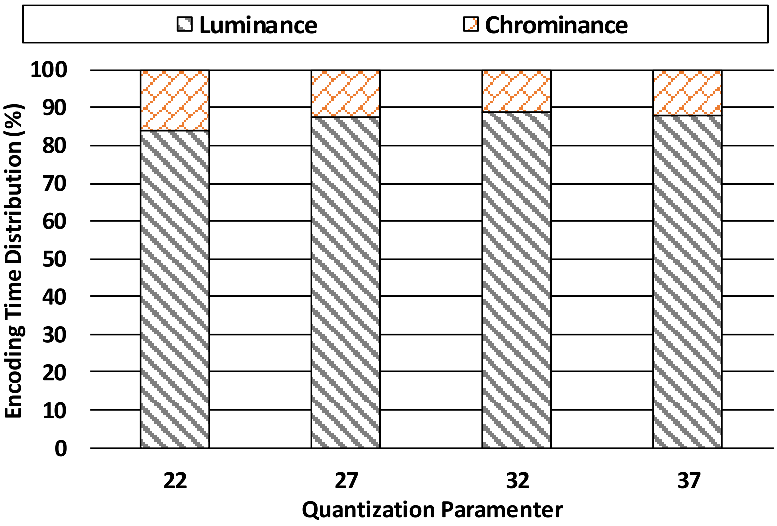


Figure 14 – Encoding time distribution for luminance and chrominance components considering the QP values defined in CTC.

The luminance coding has the highest computational cost in VTM intra-frame coding, obtaining a maximum and minimum of 89% and 84% of the total encoding time when considering QP=32 and QP=22, respectively. It occurs because the chrominance component is subsampled and assesses only eight prediction modes applying DCT-II or TSM, and LFNST for residual coding. In contrast, luminance still evaluates AIP, MRL, MIP, ISP, and MTS coding tools. On average, luminance coding demands 87% of the computational effort of the VTM encoder in the all-intra scenario.

The next evaluations consider the encoding time and usage distribution of VVC intra-frame prediction focusing on analyzing the luminance block sizes and coding tools.

5.4. Block Size Analysis

Figure 15 and Figure 16 presents the encoding time distribution and the usage distribution for each available luminance block size considering the QP corner cases, respectively. The blue and orange bars denote the results for QP=22 and QP=37, respectively. The x-axis represents each block size, ordered from the largest to the smallest one, ranging from 64×64 to 4×4 samples. Note that there are no rectangular-shaped blocks with a width or height of 64 samples since MTT partitioning is only performed over 32×32 blocks or smaller, regarding I-slices (HUANG et al., 2021).

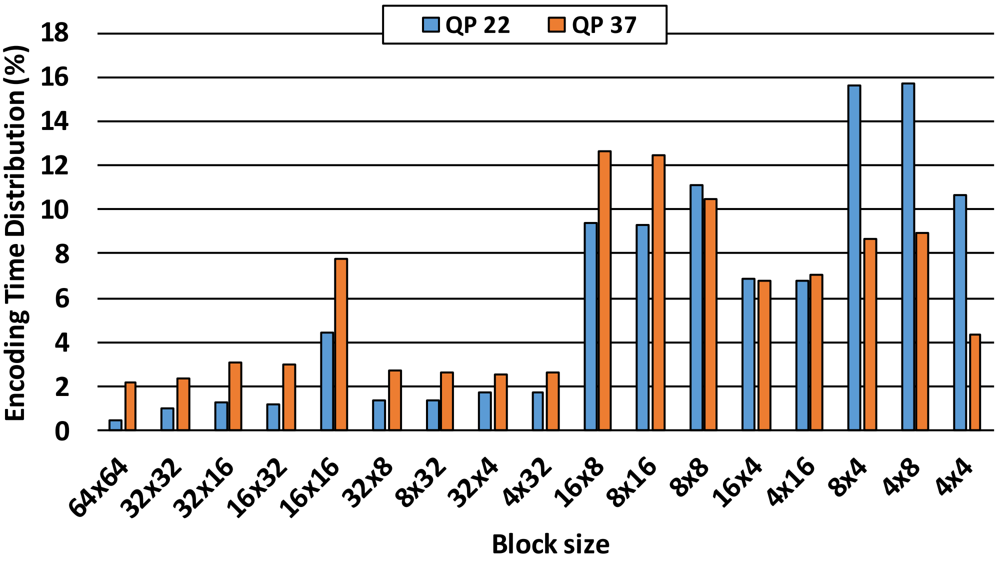


Figure 15 – Encoding time distribution for luminance block sizes considering QP=22 and QP=37.

For both QPs evaluated in Figure 15, the most time-consuming blocks have sizes of 16×16, 16×8, or smaller. However, the QP variation produces different encoding time distribution for each block size; lower QPs concentrate the encoding effort in the block sizes with smaller areas, whereas higher QPs have a more heterogeneous complexity distribution. For instance, the total encoding time of block sizes smaller than or equal to 8×8 is 66.8% for QP=22 and 46.2% for QP=37.

Analyzing the average occurrence of each available block size in Figure 16, one can notice that the block size selection is highly dependent on the QP value; higher and lower QP values imply selecting larger and smaller block sizes, respectively.

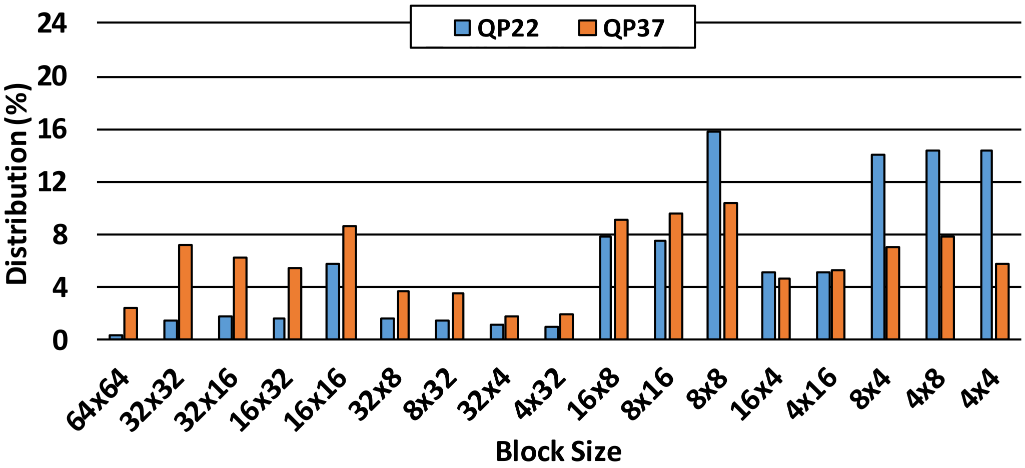


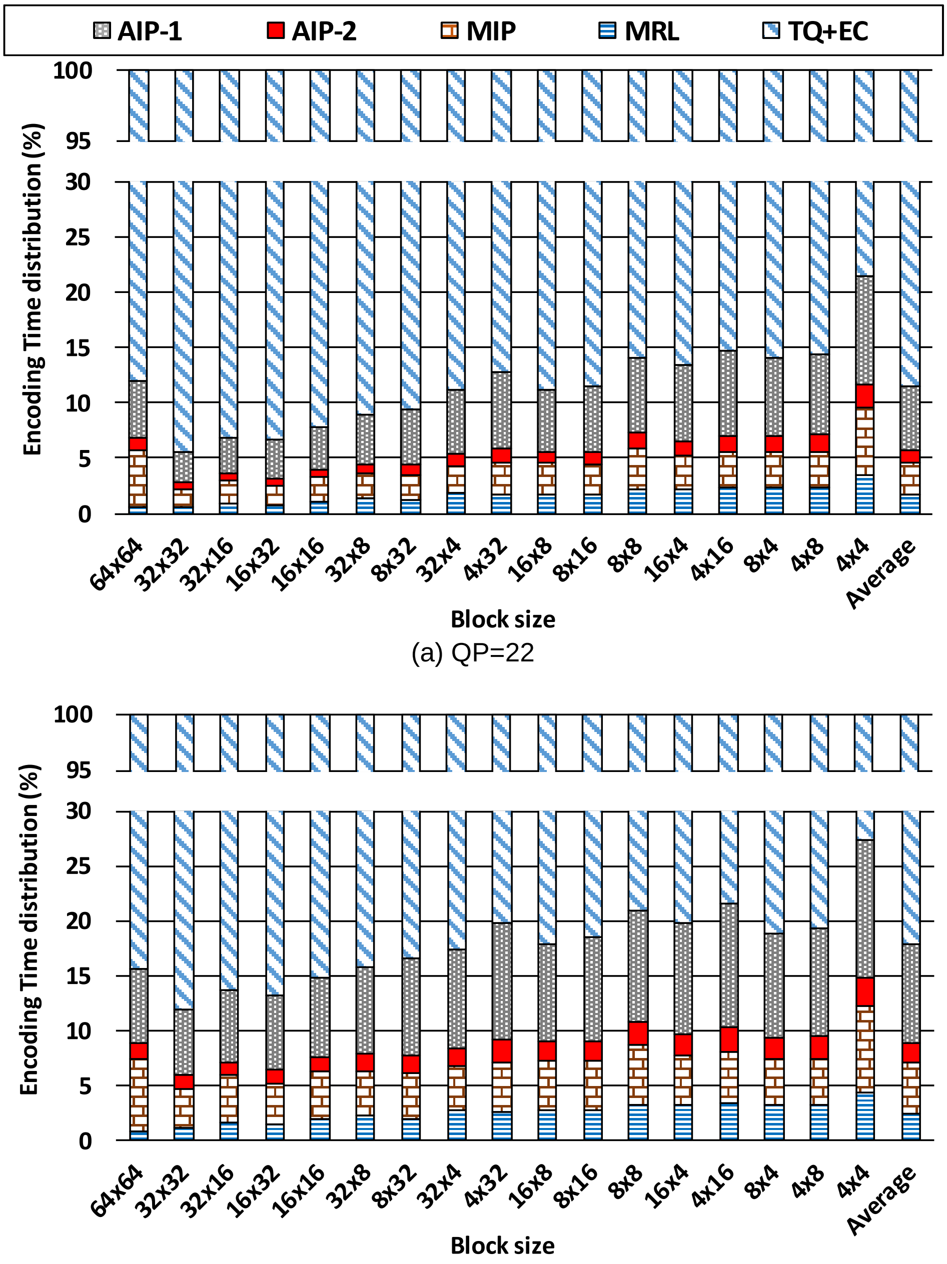
Figure 16 – Usage Distribution for luminance block sizes considering QP=22 and QP=37.

The block selection for QP=22 concentrates in 16×8 samples or smaller, with 84% of the occurrences. Blocks larger than 16×8 occur less than 2% of the times, except for the 16×16 block that occurs 5.7% of the times. A more heterogeneous distribution is noticed with QP=37, where blocks from 16×8 samples or smaller occur 59.3% of the times, showing a reduced usage compared to QP=22. Hence, the percentage of blocks larger than 16×8 has increased to a total of 40.7%. This distribution happens because low QP values retain more image details, producing more heterogeneous regions, which are better encoded with smaller blocks. In contrast, to raise the compression rate, high QP values attenuate the image details, producing more homogeneous regions that are better encoded with larger blocks.

5.5. Encoding Mode Analysis

Figure 17 displays the encoding time distribution of the encoding intra prediction steps according to the block size and the QP corner cases. This analysis considers AIP-1, AIP-2, MRL, and MIP as prediction steps, and TQ+EC as the residual coding flow regarding transform, quantization, and entropy coding. ISP and MPM have negligible processing time since they derive the prediction modes from predefined lists; thus, these tools were not considered in this analysis as a prediction step.

The residual coding (TQ+EC) is the most time-demanding process for all block sizes and both corner QPs; all the other steps together are responsible for less than 30% of the total encoding time in all cases. Then, Figure 17 omits part of the residual coding distribution to visualize better the other steps. Figure 17(a) presents the encoding time distribution with QP=22 and Figure 17(b) with QP=37. Comparing both graphs, one can conclude that the computational burden of residual coding decreases for higher QPs; in this case, the prediction tools represent a higher percentage of the encoding time. AIP-1 and MIP are the prediction tools that concentrate the highest encoding effort in both cases, with a maximum of 8.9% and 4.7% of the total encoding time (QP=37). MRL and AIP-2 demand, together, less than 4.5% of the encoding time, on average, in both cases.



(a) QP=22

(b) QP=37

Figure 17 – Encoding time distribution for the intra-frame prediction tools considering (a) QP=22 and (b) QP=37.

The high computational cost of the residual coding is mainly noticed in the transform and quantization steps, demonstrating that MTS and LFNST evaluations increased the encoding complexity of the transform process. Additionally, VVC introduces several new intra prediction modes to be evaluated by the residual coding. Although ISP presents a negligible processing time in the prediction step, this tool can add up to 48 prediction modes (16 modes for each LFNST index) in the RD-list to be evaluated by the residual coding, contributing to this high complexity. Another conclusion when observing Figure 17 is as smaller is the block size, as higher tends to be the percentual effort spent in the prediction steps. This occurs mainly because of the relation of the available encoding options and the number of samples per block size. This relation tends to concentrate the prediction effort in the smaller block sizes.

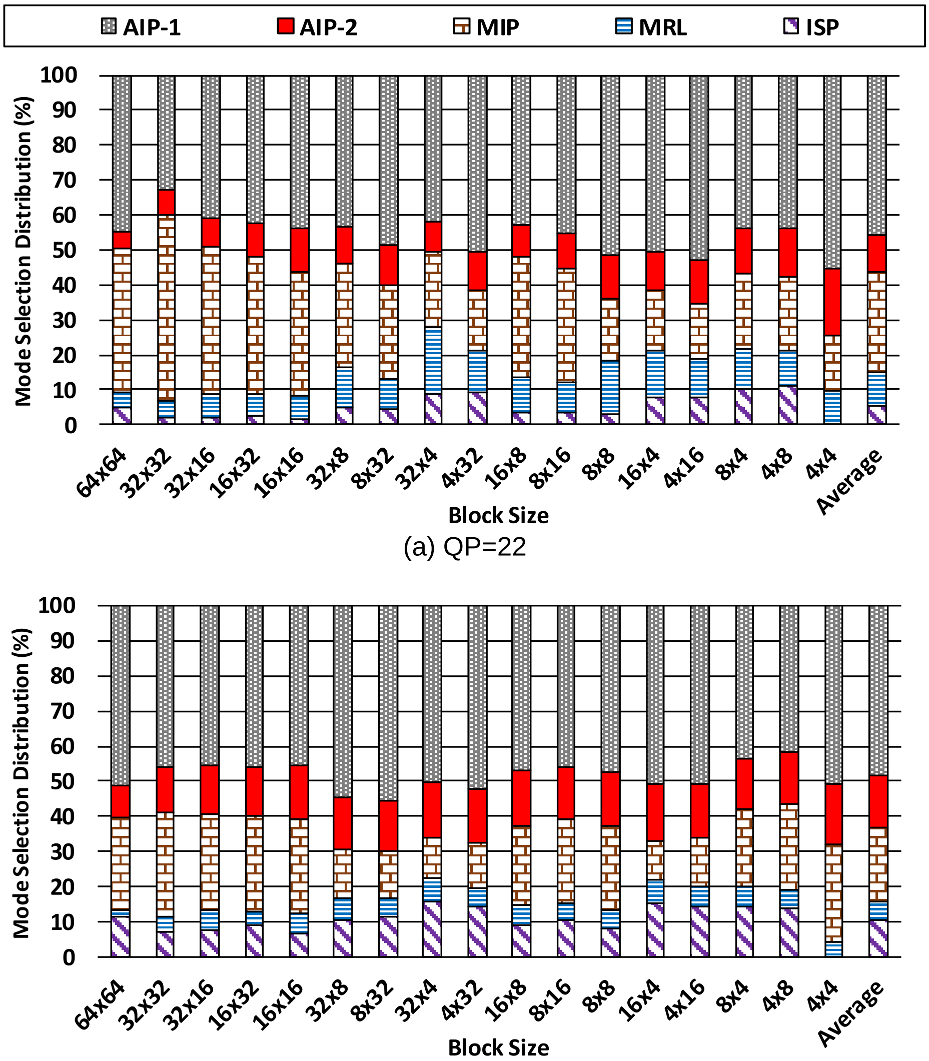
This analysis showed that the residual coding of VVC intra-frame prediction had significantly raised its computational effort, presenting the highest encoding time for all assessed cases. It occurs because, for each prediction mode in the RD-list, the residual coding flow is done several times, considering new prediction modes and different combinations of primary and secondary transforms.

Another interesting analysis to understand the encoder decisions is related to the encoder modes distribution using intra-frame prediction. The next analysis considers the prediction mode selection distribution among the available intra-frame prediction modes. Figure 18 shows this analysis considering AIP-1, AIP-2, MRL, MIP, and ISP prediction modes. For both QP values in Figure 18(a) and Figure 18(b), AIP-1 is the most used mode, followed by MIP. MRL is more used than ISP for lower QPs, but this order is inverted for higher QPs.

For all evaluated QPs, more than 45%, 20%, and 10% of the cases use AIP-1, MIP, and AIP-2, respectively. The QP value has a different impact on the encoding mode distribution; the higher the QP value, the higher the use of AIP-1, AIP-2, and ISP tools. Naturally, MIP and MRL present the opposite behavior.

The encoding tools also have different usage behavior, considering the block sizes. The higher the block size, the higher the use of MIP, especially for lower QPs. Considering the QP=22, the MIP is even more used than the AIP-1 for some blocks larger than 16×16 samples. AIP-2 and MRL tend to be less used for larger block sizes, mainly for lower QPs. ISP also follows this trend but with a less linear behavior.

This analysis demonstrates that although VVC brings new tools for intra-frame prediction, the HEVC intra-frame prediction modes (AIP-1) remain used a lot, providing high coding efficiency for several cases. Nevertheless, the new VVC intra-frame coding tools are essential to increase this encoding performance since these tools are selected more than 51.5% of the times, on average.



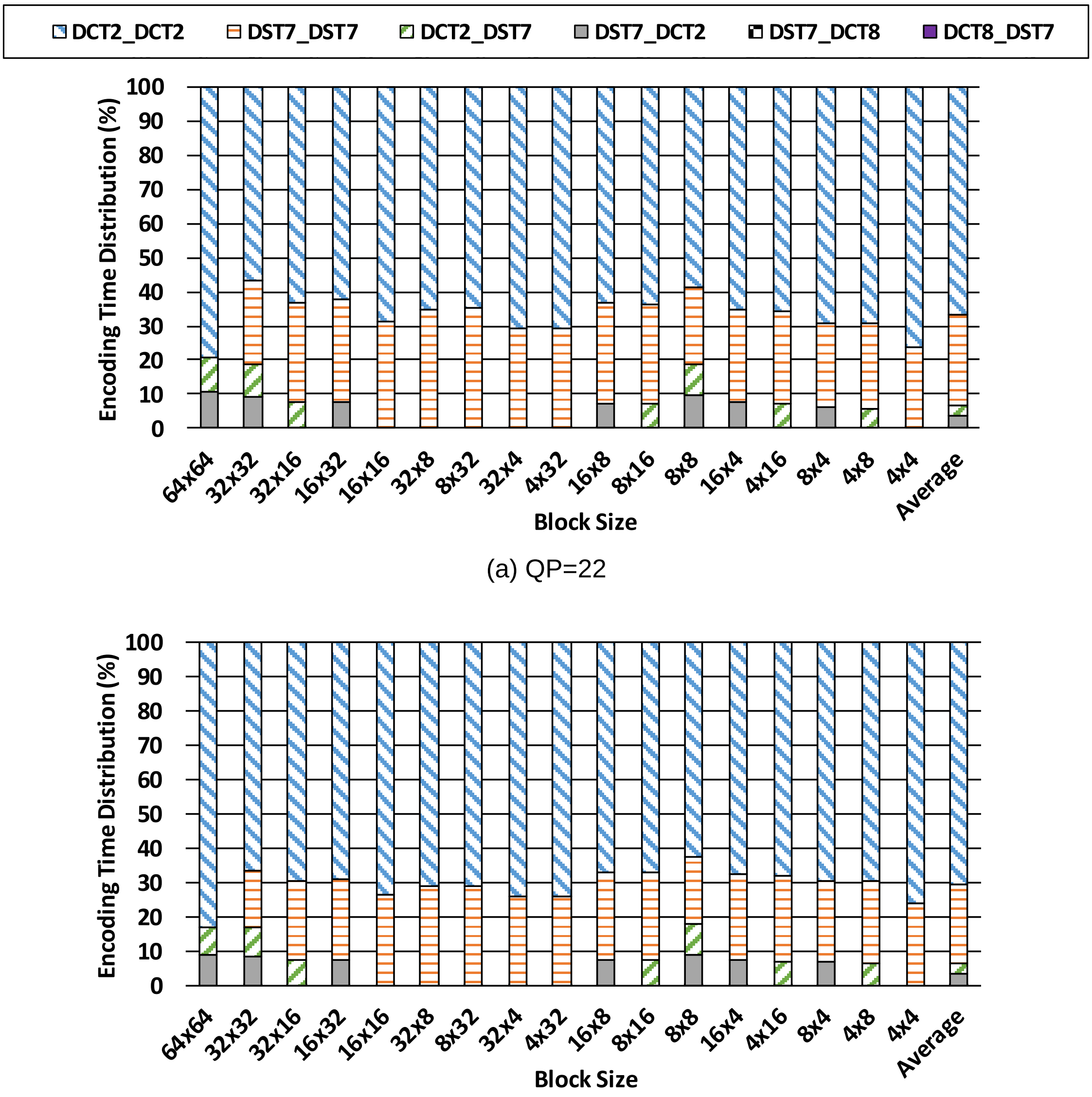
(a) QP=22

(b) QP=37

Figure 18 – Mode selection distribution for intra-frame prediction tools considering (a) QP=22 and (b) QP=37.

5.6. Encoding Transform Analysis

Since the transform step demand a high computational effort, the following analysis exhibits the encoding time distribution of primary transform combinations. Figure 19 displays the average encoding time for multiple transform selection regarding each block size and QP corner cases. This analysis considers six horizontal and vertical transform combinations: DCT-II for both directions (DCT2_DCT2), DST-VII for both directions (DST7_DST7), DCT-II for vertical and DST-VII for horizontal direction (DCT2_DST7), DST-VII for horizontal and DCT-II for vertical direction (DST7_DCT2), DST-VII for horizontal and DCT-VIII for vertical direction (DST7_DCT8), and DCT-VIII for horizontal and DST-VII for vertical direction (DCT8_DST7). Since VTM evaluates DCT2_DCT2 and TSM in the same execution flow (MTS index 0), the encoding time of DCT2_DCT2 also encompasses the encoding time of TSM.



(a) QP=22

(b) QP=37

Figure 19 – Encoding time distribution for multiple transform selection considering (a) QP=22 and (b) QP=37.

It is important to highlight that the DCT-II and DST-VII transforms can be combined only for the ISP predicted blocks and LFNST index 0 (i.e., without secondary transform), as discussed in Section 3.2.6 of this Thesis. In this case, DST-VII is implicitly applied in the horizontal, vertical or both directions if the block width, height or both have between 4 and 16 samples (inclusive); otherwise, DCT-II is applied. Even though MTS allows DCT-VIII for horizontal and vertical directions (DCT8_DCT8), this transform combination has not been performed for any block size. Besides, note that combinations of DST-VII and DCT-VIII for vertical and horizontal directions (DST7_DCT8 and DCT8_DST7) have low representativeness in the encoding time distribution, and it is impossible to find them in the graphs since these transforms are responsible for less than 0.1% of the encoding effort, on average.

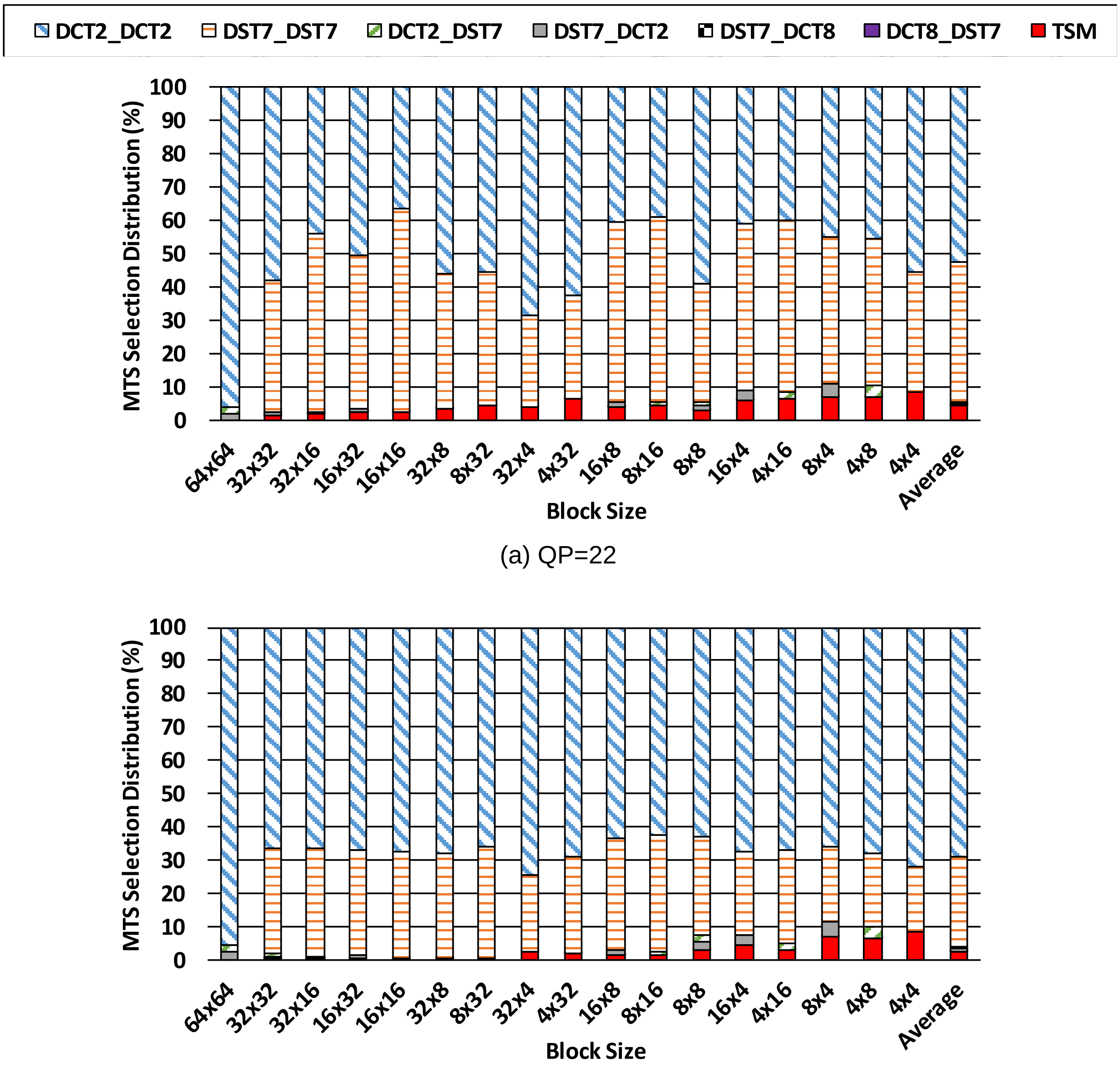
Figure 19(a) and Figure 19(b) show that DCT2_DCT2 is the most time-consuming transform operation for all block sizes and both QPs. DST7_DST7 shows the second-highest encoding effort. On average, for the corner QPs, DCT2_DCT2 and DST7_DST7 represent about 70% and 25% of the encoding effort, respectively. The remaining transform combinations have less than 6.8% of the encoding time. For higher QPs, the DCT2_DCT2 combination tends to have a relative encoding time slightly higher than the other transforms. On the other side, the use of transforms according to the block size does not correlate significantly.

DCT2_DCT2 presents the highest encoding time because this process in VTM evaluates DCT-II and TSM without secondary transform (i.e., LFNST index 0), and DCT-II with secondary transform-sets one and two (i.e., LFNST indexes 1 and 2), whereas LFNST is not performed for the remaining transform combinations. The prediction modes that obtained high RD-cost using DCT2_DCT2 are discarded for the next evaluations of transform combinations. Besides, VTM implements fast decisions based on the obtained RD-cost by applying DCT-II/TSM to evaluate the next transform combinations conditionally.

The next analysis was done to evaluate the usage distribution of multiple transforms. The primary transforms were evaluated regarding each available transform block size and the QP corner cases. Figure 20 presents this evaluation, considering the same transform combinations of the encoding time distribution shown in Figure 19. This analysis considers a TSM computation separated from DCT2_DCT2.

DCT2_DCT2 and DST7_DST7 are the most selected transform combinations for both QPs, obtaining together more than 94% of the usage distribution, on average. Another observation is that the higher the QP value, the higher is the use of DCT2_DCT2, and the opposite behavior is noticed for DST7_DST7. For QP=37, DCT2_DCT2 is the most used transform combination for all block sizes. However, for QP=22, DST7_DST7 is the most selected transform combination for block sizes 32×16, 16×16, 16×8, 8×16, 16×4, and 4×16.

The higher the QP value, the lower the use of TSM, DST7_DCT8, and DCT8_DST7 combinations. TSM is used 4.2% and 2.5% in QP=22 and QP=37, respectively, and DST7_DCT8 and DCT8_DST7 are used less than 0.1% in both cases. DCT2_DST7 and DST7_DCT2 have the opposite behavior, slightly increasing from 1.4% (QP=22) to 1.6% (QP=37). This occurs because these combinations are only allowed when ISP is used, which is more used for higher QPs. According to the block size, the behavior of the transforms does not present any observable trend for most of the combinations, considering the two QP values. Only TSM has a clear trend to be more selected for smaller block sizes for both QPs.



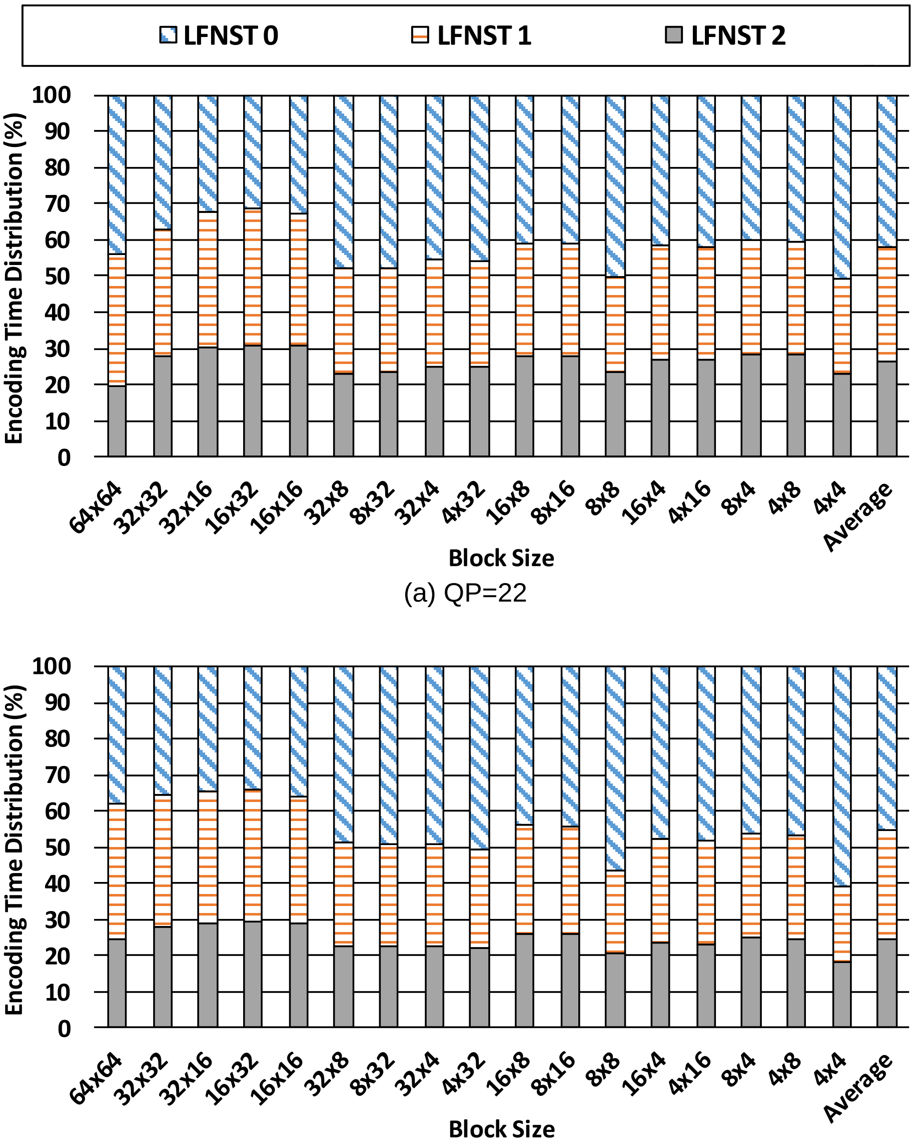
(a) QP=22

(b) QP=37

Figure 20 – Selection distribution of multiple primary transforms considering (a) QP=22 and (b) QP=37.

The low usage of transform combinations using DCT-VIII matrices is justified because MTS was designed without considering a secondary transform operation for DCT-II. The current VTM implementation inserts LFNST into the encoder resulting in satisfactory rate-distortion performance for most cases by evaluating only the DCT-II (with and without LFNST), TSM, and DST-VII transforms; therefore, justifying the low usage of transform combinations using DCT-VIII matrices.

Figure 21 illustrates the encoding time distribution among the LFNST encoding possibilities according to the block sizes and QP corner cases.



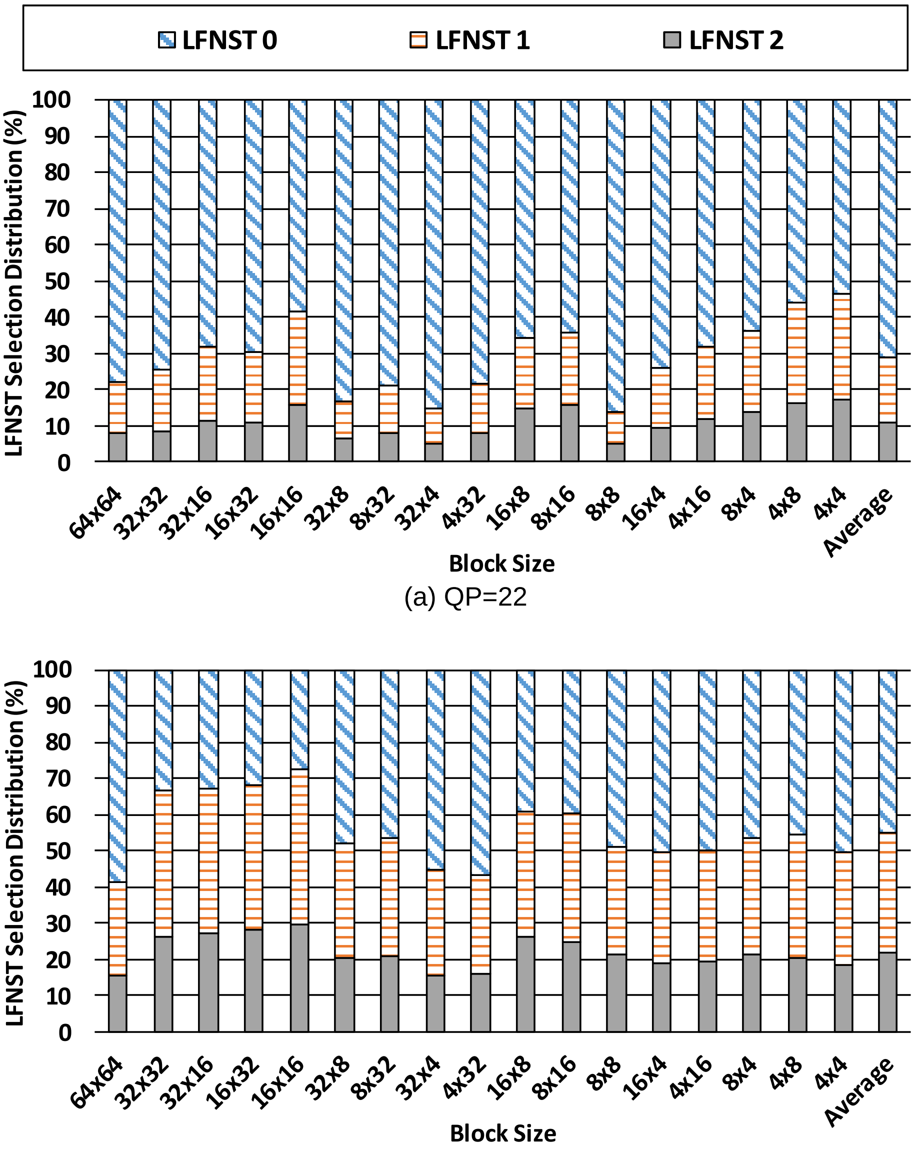
(a) QP=22

(b) QP=37

Figure 21 – Encoding time distribution of LFNST considering (a) QP=22 and (b) QP=37.

LFNST 0 refers to the residual coding without applying the secondary transform (i.e., only primary transform is applied), and LFNST 1 and LFNST 2 represent the use of secondary transform-sets one and two, respectively. For both QPs, LFNST 0 represents the highest encoding effort, followed by LFNST 1 and LFNST 2. It occurs because the VTM encoder generates and processes RD-list with DCT-II/TSM during the LFNST 0 evaluation. When LFNST 1 and LFNST 2 are processed, only DCT-II is evaluated, and the RD-list is derived from LFNST 0 processing. The VTM encoder follows a sequential evaluation, where LFNST 0 is always evaluated. In contrast, LFNST 1 and LFNST 2 are conditionally evaluated based on the obtained RD-cost by performing LFNST 0 and the generated Coded Block Flag (CBF) of the previous evaluation that signals if the block has any significant (i.e., non-zero) coefficients. Nevertheless, considering the total encoding time of both secondary transform evaluations (LFNST 1 and LFNST 2), these operations represent more than 55% of the coding time, on average. Finally, observing Figure 21, one can conclude that the computational effort of LFNST does not directly correlate with the used QP value and block size.

Figure 22 depicts the selection distribution of the secondary transform, also considering each available block size for the two corner QPs.



(a) QP=22

(b) QP=37

Figure 22 – LFNST selection distribution considering (a) QP=22 and (b) QP=37.

The secondary transform (LFNST 1 and 2) is less frequently used for lower QPs, being used 29.1% of the times with QP=22 and 55.3% of the times with QP=37, on average. This occurs because the LFNST is applied only for the DCT2_DCT2 transform combination; then, this usage distribution follows the same trend presented in the previous analysis, where DCT2_DCT2 is also more used for QP=37. Another important observation is that LFNST 1 is higher used than LFNST 2 for all evaluated cases. Figure 22 displays that there is no clear correlation between the use of the secondary transform and the block size variation.

5.7. Rate-Distortion and Computational Effort of VVC Intra-Frame Coding Tools

This section presents a rate-distortion and computational effort evaluation of the new block partitioning structure with binary and ternary partitions and the novel intra-frame coding tools when running in the VTM 10.0 under all-intra configuration. This analysis shows the impact of each block partition structure and intra-frame coding tool by removing it from the encoding flow. The evaluation measures the compression efficiency and computational effort through BDBR and encoding time saving, respectively.

Table 2 presents the BDBR increase and Encoding Time Saving (ETS) results when removing BT, TT, or both partitioning structures of VVC intra-frame coding for each class of CTC test sequences. These partitions are removed for both luminance and chrominance coding trees.

Table 2 – Compression efficiency and timesaving results when removing BT, TT, or both partitioning structures.

Class	**BT less**		**TT less**		**BT+TT less**	
	BDBR	**ETS**	**BDBR**	**ETS**	**BDBR**	**ETS**
A1	4.4%	72.9%	0.7%	42.1%	12.4%	90.9%
A2	4.9%	78.7%	1.0%	48.6%	16.0%	94.5%
B	5.8%	77.6%	1.1%	48.4%	22.1%	94.7%
C	8.5%	79.7%	1.6%	51.9%	36.2%	95.5%
D	6.7%	77.5%	1.3%	51.4%	30.7%	93.1%
E	8.6%	76.0%	1.8%	48.1%	39.3%	92.9%
Avg.	**6.5%**	**77.1%**	**1.2%**	**48.4%**	**26.1%**	**93.6%**

This evaluation allows us to assess the influence of the new partitions in the QTMT structure. On average, when the BT partitioning is disabled of the VTM encoder

(BT less in Table 2), the encoding time is reduced by 77.1% at the cost of a BDBR increase of 6.5%. When disabling the TT partitioning (TT less in Table 2), BDBR increases by 1.2%, and the encoding time is reduced by 48.4%. Disabling both BT and TT partitions (BT+TT less in Table 2), i.e., when only QT partitions are available, the encoding time is decreased by 93.6%, but, as a drawback, the BDBR is increased by 26.1%, on average.

Although the impact in BDBR is high for all classes of test sequences, the impact of disabling BT and TT partitions is more prominent for video resolutions lower than 3840×2160 (Classes A1 and A2). It is justified because lower resolutions are better encoded with smaller block sizes, and BT and TT partition structures can enable different block sizes and shapes, providing a higher compression performance.

Table 3 presents BDBR and ETS results when removing horizontal (BTH and TTH) and vertical (BTV and TTV) partitions. This analysis displays that both horizontal and vertical partitions provide similar average results of TS and BDBR increase. The horizontal partition removal provides 79.9% of TS with a 5.3% increase in BDBR; disabling the vertical partitions reduces about 79.5% of the encoding time with a 5.5% BDBR increase.

Table 3 – Compression efficiency and timesaving results when removing the horizontal or vertical partitions.

Class	**Horizontal less**		**Vertical less**	
	BDBR	**ETS**	**BDBR**	**ETS**
A1	3.8%	74.7%	3.3%	74.2%
A2	3.7%	80.4%	4.0%	80.0%
B	5.6%	81.0%	4.3%	80.0%
C	6.5%	83.2%	6.5%	82.6%
D	5.7%	81.9%	6.2%	80.9%
E	6.6%	78.4%	8.6%	79.0%
Avg.	**5.3%**	**79.9%**	**5.5%**	**79.5%**

These results presented a similar behavior to the results obtained in Table 2, demonstrating the high coding efficiency provided by horizontal and vertical partitions for all classes of video resolutions; however, the impact in BDBR of disabling horizontal and vertical partitions is also more noticeable for lower video resolutions. These analyses show that the QTMT partitioning structure provides significant compression performance gains while raising the VVC intra-frame coding complexity expressively.

Table 4 presents the average results of BDBR increase and encoding time savings when removing each intra-frame coding tool from the VTM intra-frame coding flow. The highest timesaving results are obtained when disabling LFNST or ISP tools. In contrast, the highest encoding efficiency impacts are noticed when disabling the residual coding tools LFNST or MTS, increasing BDBR by 1.2%.

Table 4 – Compression efficiency and timesaving results when removing each VVC intra-frame coding tool

Class	AIP-2 less		MRL less		MIP less	
	BDBR	**ETS**	**BDBR**	**ETS**	**BDBR**	**ETS**
A1	0.6%	-0.7%	0.1%	0.8%	1.0%	10.3%
A2	0.6%	1.3%	0.2%	0.4%	0.6%	10.7%
B	0.7%	1.4%	0.4%	-0.1%	0.5%	11.4%
C	0.9%	1.6%	0.7%	0.8%	0.5%	13.0%
D	0.8%	0.4%	0.2%	0.9%	0.6%	11.7%
E	1.4%	0.2%	0.3%	-0.1%	0.7%	10.5%
Avg.	**0.8%**	**0.7%**	**0.3%**	**0.5%**	**0.6%**	**11.3%**

Class	ISP less		MTS less		LFNST less	
	BDBR	**ETS**	**BDBR**	**ETS**	**BDBR**	**ETS**
A1	0.1%	13.0%	1.4%	10.3%	1.8%	24.8%
A2	0.3%	14.2%	1.4%	13.9%	0.7%	28.7%
B	0.4%	15.1%	1.4%	14.3%	1.0%	26.7%
C	0.7%	17.7%	0.9%	15.7%	1.4%	27.1%
D	0.6%	15.9%	0.7%	15.2%	1.1%	25.4%
E	0.8%	15.1%	1.4%	14.0%	1.5%	25.5%
Avg.	**0.5%**	**15.2%**	**1.2%**	**13.9%**	**1.2%**	**26.4%**

Disable AIP-2 and MRL produce few gains in encoding time reduction (less than 1%). However, while removing the MRL represents the smallest impact in the coding efficiency, when removing the AIP-2, the BDBR is increased by almost 1%, representing the highest BDBR increase considering the prediction tools. When MIP is disabled, the encoding time is reduced by 11.3% at the cost of a BDBR increase of 0.6%.

The highest BDBR impacts of AIP-2, MRL, and MIP prediction tools are 1.4% (Class E – 1280×720 pixels), 0.7% (Class C – 416×240 pixels, and 1.0% (Class A1 - 3840×2160 pixels), respectively. ISP presents the highest and the lowest BDBR impact for Class E and Class A1/A2, respectively. MTS and LFNST transform tools present similar behavior between the classes of video sequences, except in classes C and D, where the MTS coding tool obtained a lower BDBR impact compared to the others.

This evaluation showed that each new VVC intra-frame prediction tool contributes to increasing the coding efficiency. However, this efficiency comes at the cost of a high computational effort, especially for the residual coding, which is performed several times to choose the best combination of prediction mode, primary transform, and secondary transform.

5.8. General Discussion

All new intra-frame coding tools were added to the VVC specification because they improve the encoding efficiency. Therefore, for scenarios requiring high coding efficiency these tools cannot be directly removed from the encoding flow to reduce the complexity. However, based on the analyses presented in previous sections, several ideas and conclusions can be taken to elaborate efficient timesaving solutions for VVC intra-frame prediction beyond those solutions already inserted in the VTM or proposed in the literature.

Firstly, encoding small blocks such as 4×4, 4×8, 8×4, and 8×8 takes more time compared to encoding larger block sizes, independent of the quantization scenario. Moreover, the selection of small blocks decreases according to the QP increase. In contrast, larger blocks are less frequently selected with low QPs. However, in this case, the encoding time distribution presents slight variations among the evaluated QPs. Therefore, QP can be considered when designing a timesaving solution by adaptively limiting the depth and skipping the top level of the QTMT structure. Since the QTMT structure comprises three partitioning structures, some approaches for time reduction solutions can be considered, including the following predictions: quadtree depth, MTT depth, when using BT, TT, or both, and when using MTT horizontal or vertical partitioning.

The analyses of the intra-frame coding flow demonstrated TQ+EC is the most time-consuming module regardless of the quantization scenario. In this case, solutions that can reduce the number of prediction modes evaluated in the TQ+EC flow (i.e., reduce the RD-list) should be considered to provide more impressive encoding time reduction results. A more limited encoding time reduction can be obtained if considering the prediction tools. In this case, AIP-1 is the most time-demanding tool and simplifying the RMD search step can also save the encoding time.

The residual coding is responsible for a considerable encoding effort to evaluate each intra-frame prediction mode through the TQ+EC flow for all available block sizes. This computational burden is mainly due to the evaluations of primary and secondary transforms. Regarding the primary transform, DCT2_DCT2 and DST7_DST7 are the most complex transform operations for all evaluated quantization scenarios. While DST7_DST7 is more selected for a low QP (reducing the usage of DCT2_DCT2), for a high QP, the use of DST7_DST7 decreases, and the use of DCT2_DCT2 increases expressively. For the secondary transform, the three possibilities of LFNST encoding take similar encoding time for evaluated QPs. However, the analysis demonstrated that LFNST 1 and LFNST 2 are less frequently selected for lower QPs, whereas the opposite happens for higher QPs. In this case, solutions considering the encoding context can be developed to reduce the number of transform combinations evaluated in the intra-frame coding flow, including predicting the primary transform combination and when to use or not the secondary transform.

6. FAST MULTI-TYPE TREE DECISION SCHEME FOR LUMINANCE BLOCKS

This chapter presents a fast decision scheme for the MTT block partitioning structure based on statistical analysis for luminance blocks. As the conventional angular intra-frame prediction modes (AIP in Section 3.2.1) and ISP mode tend to indicate the texture direction of the block, the best mode of AIP and ISP sub-partitions direction (ISP_{hor} and ISP_{ver}) can be an effective predictor for the MTT partitioning decision. Furthermore, the features of the block samples can be explored to decide the direction of the binary and ternary split since MTT tends to divide the block into regions sharing more similar sample values for providing accurate predictions. For this purpose, we design a scheme composed of two strategies that explore the correlation of the intra-frame prediction modes and samples of the current CU for deciding the split direction of binary and ternary partitions. Based on this information, our scheme can avoid unnecessary evaluations of binary and ternary partitions, reducing the encoding time with similar coding efficiency. This solution and original results were published in (SALDANHA et al., 2020b).

6.1. Motivational Analysis

Figure 23 presents the luminance CU partitions for the first frame of the BasketballPass video sequence, which was encoded with QP 37 and all-intra configuration; three blocks with horizontal splits are detached in blue boxes, and three blocks with vertical splits are highlighted in red boxes. To correlate the luminance samples for deciding the direction of binary and ternary split, we defined two variance values for a given luminance block: (i) var_{hor} refers to the sum of the variance values of upper and lower partitions, considering the current block is horizontally subdivided into two equal-sized regions, and (ii) var_{ver} indicates the sum of the variance values of the left and right partitions for a current block vertically subdivided into two equal-sized regions. On the one hand, Figure 23 shows that when the horizontal CU partitioning (blue boxes) occurs, var_{hor} tends to be smaller than var_{ver}. On the other hand, when var_{ver} is smaller than var_{hor} (red boxes), the CU tends to split vertically. Thus, var_{hor} and var_{ver} can be employed to predict the direction of binary and ternary split, avoiding unnecessary MTT evaluations.

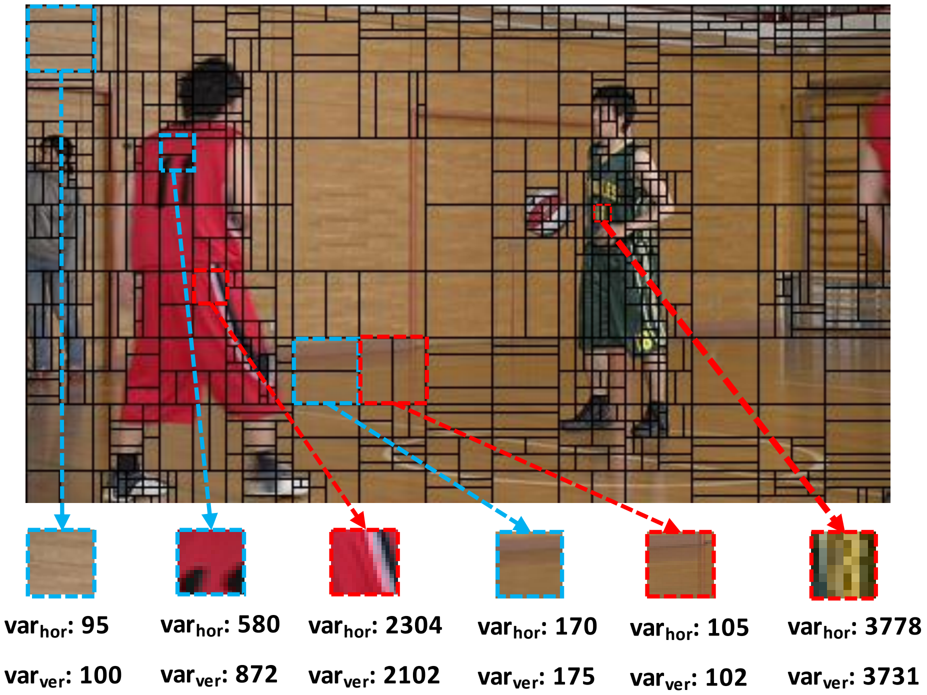

$\mathrm{var}_{\mathrm{hor}}$: 95
$\mathrm{var}_{\mathrm{ver}}$: 100
$\mathrm{var}_{\mathrm{hor}}$: 580
$\mathrm{var}_{\mathrm{ver}}$: 872
$\mathrm{var}_{\mathrm{hor}}$: 2304
$\mathrm{var}_{\mathrm{ver}}$: 2102
$\mathrm{var}_{\mathrm{hor}}$: 170
$\mathrm{var}_{\mathrm{ver}}$: 175
$\mathrm{var}_{\mathrm{hor}}$: 105
$\mathrm{var}_{\mathrm{ver}}$: 102
$\mathrm{var}_{\mathrm{hor}}$: 3778
$\mathrm{var}_{\mathrm{ver}}$: 3731

Figure 23 – CU partitioning of BasketballPass video sequence with variance values of the highlighted blocks.

Additionally, we correlate the encoding context through the AIP modes when it was chosen as the best mode for the current CU to further improve the performance of this strategy. Therefore, motivated by the work of Fu et al. (2019a), we divided the AIP modes into two categories, where AIP modes from 10 to 28 (horizontal mode ± 8) are defined as the horizontal directions ($\mathrm{AIP}_{\mathrm{hor}}$), and AIP modes from 42 to 58 (vertical mode ± 8) are specified as the vertical directions ($\mathrm{AIP}_{\mathrm{ver}}$).

Based on this analysis, we proposed a strategy called Fast Decision based on Variance (FDV). The FDV strategy identifies the texture direction according to the variance values and the best AIP mode of the current CU. Thus, if $\mathrm{var}_{\mathrm{hor}}$ is smaller than $\mathrm{var}_{\mathrm{ver}}$ and the intra-frame prediction mode is $\mathrm{AIP}_{\mathrm{hor}}$, the encoder can skip the vertical binary and ternary splitting evaluations; otherwise, if $\mathrm{var}_{\mathrm{ver}}$ is smaller than $\mathrm{var}_{\mathrm{hor}}$ and the intra prediction mode is $\mathrm{AIP}_{\mathrm{ver}}$, the encoder can skip the horizontal binary and ternary splitting.

The second strategy, named Fast Decision based on ISP (FD-ISP), correlates the ISP with the MTT structure due to the partitioning similarity. Thus, if the best mode is $\mathrm{ISP}_{\mathrm{hor}}$, the encoder can skip the vertical binary and ternary splitting evaluations; otherwise, if the best mode is $\mathrm{ISP}_{\mathrm{ver}}$, the encoder can skip horizontal binary and ternary splitting.

We evaluated seven video sequences with different characteristics under the all-intra configuration for four QP values (22, 27, 32, and 37) to analyze the accuracy of these strategies. These sequences include Campfire, CatRobot, BasketballDrive, RitualDance, RaceHorsesC, BasketballPass, and FourPeople. Figure 24 presents the accuracy of FDV, FD-ISP, and Overall, which presents the result when both strategies are jointly evaluated. Accuracy measures the ratio of correct predictions over the total number of instances evaluated. One can notice that the accuracy of FDV is higher than 92% for all cases evaluated, whereas the accuracy of FD-ISP is higher than 88%. The Overall results demonstrate the effectiveness of the proposed scheme with high accuracy of 92.1%, on average.

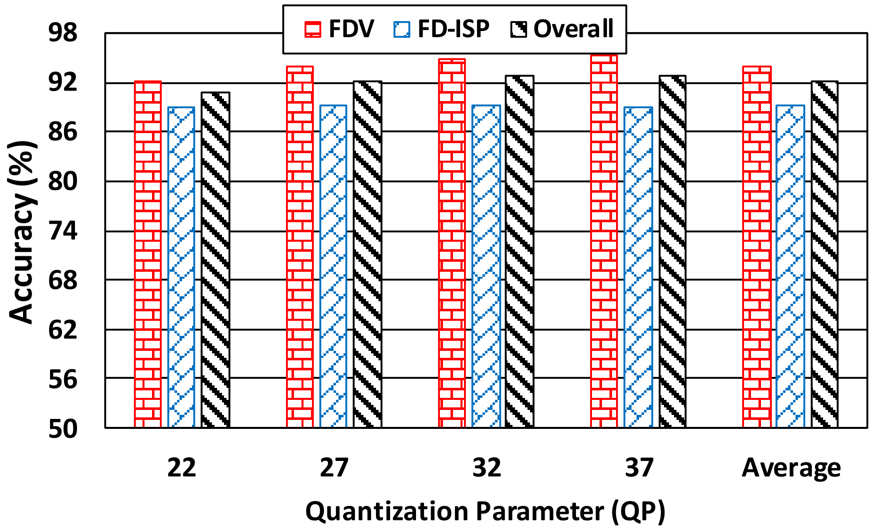


Figure 24 – Accuracy for FDV, FD-ISP, and Overall strategies.

6.2. Designed Scheme

Figure 25 shows the flowchart of the proposed fast partitioning decision scheme, which was designed based on the previous analysis. For encoding a CTU, the intra prediction and QTMT partitioning are performed sequentially. If the current CU is partitioned by QT structure, no coding simplification is performed, and the next QT depth is evaluated since the proposed scheme works only in the MTT structure. However, if the current CU is partitioned using the MTT structure with horizontal splitting, the encoder verifies the proposed fast partitioning decision FD_{hor}. When FD_{hor} is true, the current CU is classified as vertical texture direction, and the horizontal binary and ternary splitting are skipped; otherwise, no simplification is done, and the next MTT depth is evaluated. FD_{ver} is analogous to FD_{hor} but skipping the vertical splitting.

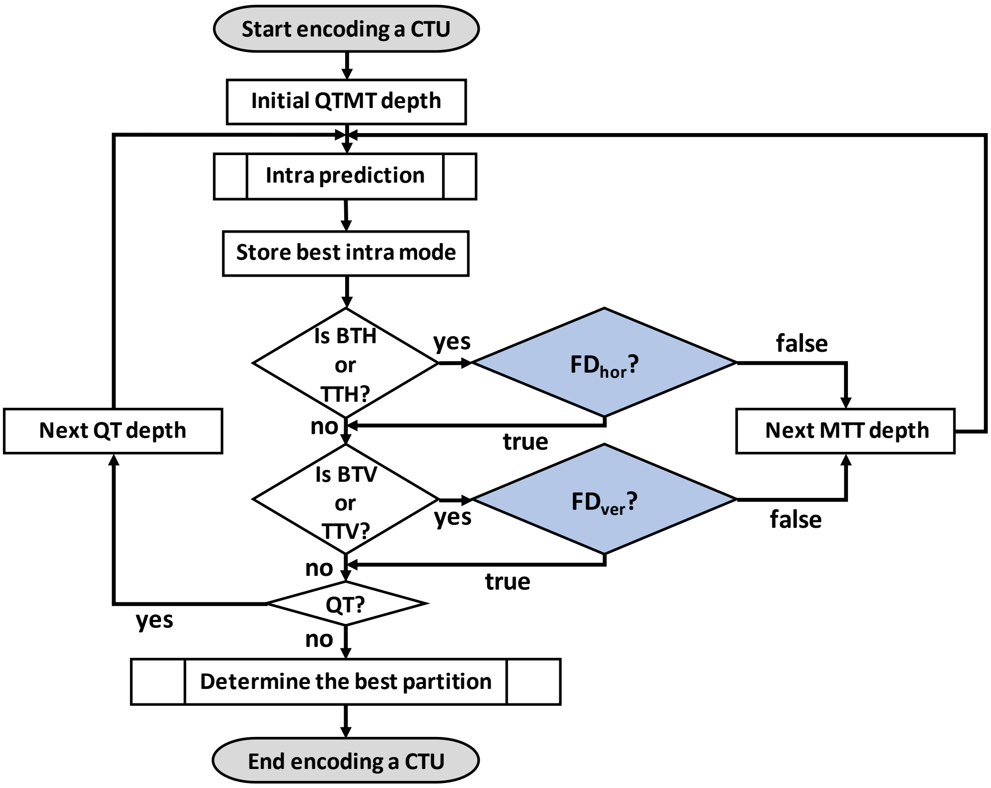


Figure 25 – Flowchart of the proposed solution.

Equation 2 defines FD$_{hor}$, which is responsible for the fast decision to skip the horizontal binary/ternary splitting evaluations. Equation 3 specifies FD$_{ver}$, which controls the fast decision to skip the vertical binary/ternary splitting.

$$FD_{hor} = \begin{cases} True, & if\ (AIP_{ver}\ and\ var_{ver} < var_{hor})\ or\ ISP_{ver} \\ False, & otherwise \end{cases} \tag{2}$$

$$FD_{ver} = \begin{cases} True, & if\ (AIP_{hor}\ and\ var_{hor} < var_{ver})\ or\ ISP_{hor} \\ False, & otherwise \end{cases} \tag{3}$$

6.3. Results and Discussion

The proposed fast partitioning decision scheme was implemented inside the VTM version 5.0 and evaluated following the JVET CTC for SDR video sequences (details in APPENDIX A). This was the first solution developed in this Thesis and it was designed and published before the VVC standardization; consequently, the published results considered a draft version of the VVC reference software. However, new experimental results using the VTM version 10.0 with all standardized coding tools also were generated for this Thesis and presented in this section. The evaluations were performed under the all-intra encoder configuration and considered the six classes of CTC video sequences. All the results presented in the next solutions of this Thesis also consider VTM version 10.0 following the CTC under all-intra encoder configuration.

Table 5 presents the results achieved with the proposed solution for VTM 10.0; the encoding efficiency was measured by the BDBR metric and encoding timesaving (ETS). The proposed solution obtained an ETS of 28.78%, on average, which varies from 8.12% to 43.23% according to the encoded video sequence. As a drawback, BDBR increases by 0.80%. These results demonstrated that our solution reduces the encoding time significantly, with a minor impact on the encoding efficiency.

The video sequences with a low selection of ISP and AIP modes, as the ParkRunning3, obtained smaller ETS since our solution takes advantage of these prediction modes to accelerate the encoder. For example, in BasketballDrive, 33.42% of the encoded CUs are predicted with ISP, whereas in ParkRunning3, ISP is used only 10.21% in the encoded CUs. However, even in this case, our solution can achieve satisfactory timesaving results with small BDBR loss.

Table 6 summarizes the comparison of the proposed solution with related works. The related works (YANG et al. 2020; FU et al.; 2019a; LEI et al. 2019) also developed fast solutions for accelerating the VVC intra coding focusing on the block partitioning structure. Yang et al. (2020) proposed a solution that achieves an average encoding time saving of 52.59% with a BDBR increase of 1.56%. Fu et al. (2019a) proposed a scheme that reduces the encoding time by 45% with a BDBR loss of 1.02%. Lei et al. (2019) proposed a method capable of reducing 40.7% of the encoding time with a BDBR increase of 0.84%. Our solution reaches lower ETS than these works but provides better BDBR results. It is important to mention that the works (FU et al.; 2019a; YANG et al. 2020; LEI et al. 2019) were evaluated in VTM 1.0, VTM 2.0, and VTM 3.0, respectively. VTM 10.0 already includes several other fast decisions, such as split cost prediction and ternary split restriction. Additionally, VTM 10.0 includes tools such as MIP and ISP that completely change the behavior of the encoder (VTM, 2020).

The work of Cui et al. (2020), implemented in VTM 5.0, provides 51.01% of ETS and 1.23% of BDBR increase. The work of Zhao et al. (2020), implemented in VTM 7.0, reaches 39.39% of ETS and 0.86% BDBR. Again, our solution achieves better BDBR results with lower ETS. However, it is necessary to note that our solution focuses only on the MTT structure, whereas three related works focus on the QT and MTT structures. Based on this fact, the proposed solution also can be combined with other techniques focusing on accelerating QT structure or other encoding modules to provide more impressive ETS results. Nevertheless, the scheme designed provides competitive results of ETS and BDBR compared to the related works.

Table 5 – Proposed Solution results for CTC evaluation under all-intra configuration.

Class	Video Sequence	BDBR	ETS
A1	Tango2	0.30%	13.53%
	FoodMarket4	0.26%	14.57%
	Campfire	0.38%	12.81%
A2	CatRobot	0.72%	21.76%
	DaylightRoad2	0.92%	29.48%
	ParkRunning3	0.15%	8.12%
B	MarketPlace	0.24%	14.47%
	RitualDance	0.54%	25.79%
	Cactus	0.78%	27.54%
	BasketballDrive	0.97%	37.46%
	BQTerrace	1.33%	38.57%
C	BasketballDrill	1.37%	27.67%
	BQMall	1.21%	43.23%
	PartyScene	0.88%	42.31%
	RaceHorsesC	0.58%	27.71%
D	BasketballPass	0.98%	37.03%
	BQSquare	1.01%	40.19%
	BlowingBubbles	0.75%	35.50%
	RaceHorses	0.62%	25.98%
E	FourPeople	1.11%	39.13%
	Johnny	1.17%	33.52%
	KristenAndSara	1.26%	36.78%
Average		**0.80%**	**28.78%**
Standard Deviation (σ)		**0.38**	**10.70**

Table 6 – Comparison of the proposed solution with related works.

Work	VTM Version	Module	BDBR	ETS
Our	10.0	MTT	0.80%	28.78%
(FU et al., 2019a)	1.0	MTT	1.02%	45.00%
(YANG et al., 2020)	2.0	QT+MTT	1.56%	52.59%
(LEI et al., 2019)	3.0	MTT	0.84%	40.70%
(CUI et al., 2020)	5.0	QT+MTT	1.23%	51.01%
(ZHAO et al., 2020)	7.0	QT+MTT	0.86%	39.39%

7. CONFIGURABLE FAST BLOCK PARTITIONING SCHEME WITH LIGHT GRADIENT BOOSTING MACHINE FOR LUMINANCE BLOCKS

The second solution presented in this Thesis is an evolution of the previously presented scheme. This solution is also focused on the luminance block partitioning decision, based on machine learning approach. Our solution considers the QTMT block partitioning as a multiple binary classification problem, where a Light Gradient Boosting Machine (LGBM) classifier is offline trained for each split type. Each classifier is responsible for deciding to perform the split type or not, avoiding the evaluation of split types that are unlikely to be chosen as the best ones. We decided to use LGBM since it presented significantly higher performance (in terms of accuracy) than a single decision tree classifier and other similar classifiers in our early stage of evaluations and tests. Besides, this classifier has a high potential for improving the performance due to the high flexibility of the hyperparameters configuration. This solution was published in (SALDANHA et al., 2021b).

7.1. Background on LGBM Classifiers

Ensemble models in machine learning combine the decisions of multiple weak learners to improve the overall system performance (NATEKIN; KNOLL, 2013), providing higher accuracy results than individual models. The two main types of ensemble approaches are bagging that creates individual classifiers for taking decisions based on the majority votes of all classifiers, and boosting, which builds the classifiers iteratively, minimizing the error of the earlier trained classifiers (NATEKIN; KNOLL, 2013).

LGBM is a gradient boosting framework developed by Microsoft researchers using tree-based learning algorithms (KE et al., 2017). Figure 26 exemplifies the LGBM training approach that builds a decision tree ensemble sequentially to minimize losses and improve the model at each iteration step. Each iteration determines a new decision tree model training concerning the error of the entire ensemble learned so far. The learning rate controls the gradient descent approach used to minimize the loss when adding trees.

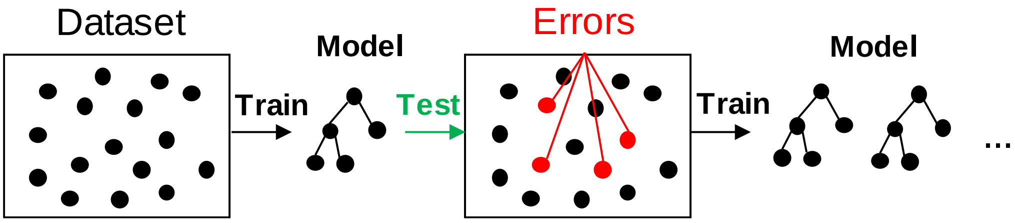


Figure 26 – LGBM training approach. Source: Adapted from (BOEHMKE, B., GREENWELL, B., 2019).

LGBM achieves a solid predictive model by combining *N* tree models ($f_1, f_2, f_3, \ldots, f_n$), and the final result described in (4) aggregates the results from each step.

$$f(x) = \sum_{n=1}^{N} f_n(x) \tag{4}$$

Unlike other tree-based learning algorithms, LGBM grows tree leaf-wise (vertically) since it can reduce the prediction loss more efficiently than algorithms that produce level-wise trees (horizontally). Moreover, conventional implementations of Gradient Boosting Machines (GBM) scan all the data instances to estimate the information gain of all possible split points, which is very time-consuming for the training process. LGBM uses Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to overcome this problem. GOSS and EFB are sampling methods for data selection, discarding some well-trained instances (small training error), and reducing the dimensionality of the features while maintaining high accuracy (KE et al., 2017).

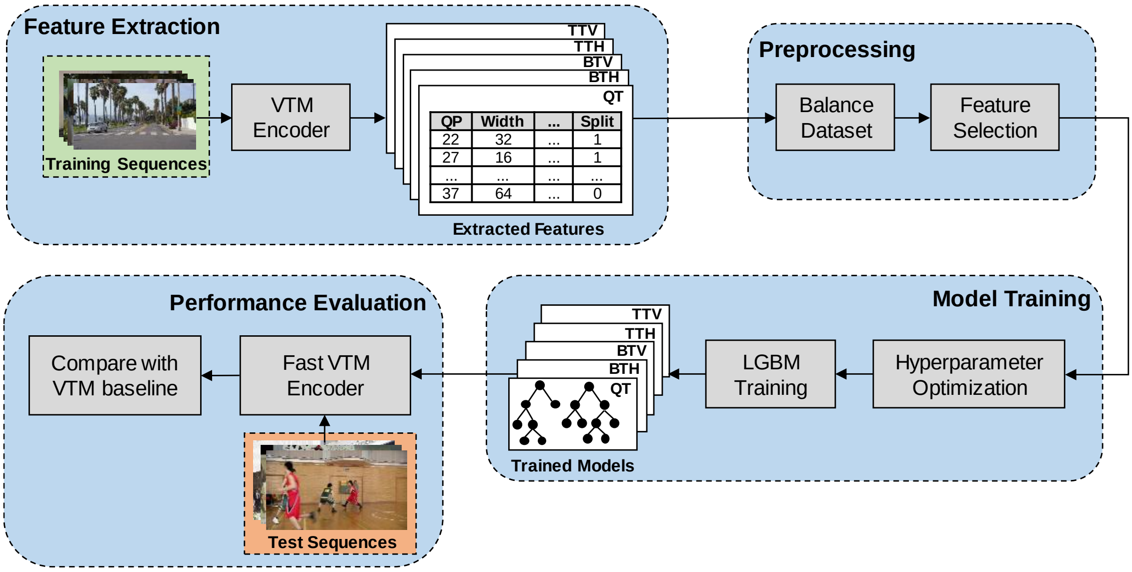
In summary, LGBM has many advantages compared to other machine-learning models, such as (i) ability to handle large-scale data, (ii) support of parallel and Graphics Processing Unit (GPU) learning, (iii) low memory usage, (iv) fast training speed, (v) simple implementation with tree-based algorithm (vi) high accuracy, and (vii) low inference time. The last three characteristics are crucial for this solution since it reduces the VVC encoding time without harming the coding efficiency.

The LGBM techniques provide a highly flexible training process to control the learning rate hyperparameters, dataset sampling, and decision tree characteristics, generating a high-efficient model when adequately optimized.

7.2. Methodology

We used data mining to discover strong correlations between the coding context and its attributes for defining machine learning models that determine when to perform a QTMT split type, saving coding time with negligible reduction in coding efficiency. Our solution divides the block partition decision into five binary classification problems instead of creating an LGBM classifier that solves the QTMT structure multiclass problem directly (more details in APPENDIX B). This approach allows the design of specialized classifiers for each split type, saving expressive encoding time while minimizing the coding efficiency loss. For this purpose, we offline trained an LGBM classifier for each split type, including QT, BTH, BTV, TTH, and TTV, and each classifier decides to skip or not the corresponding split type.

Figure 27 presents the framework used to train and implement the LGBM classifiers in the VTM encoder. A set of video sequences were selected to extract the features and train the classifiers. The VTM encoder was modified to collect several statistical data with relevant information for the CU split decision and generate the dataset of each split type. The datasets are composed of relevant features from the encoded video sequence, encoder attributes, and the split decision. The preprocessing step was performed to balance the datasets and select the most important features. The selected features are used as input for training the classifiers; this step includes hyperparameter optimization and the training of each classifier. The final step evaluates the coding efficiency and encoding timesaving using a modified VTM encoder, which incorporates the LGBM classifiers for deciding the QTMT partitioning instead of the full RDO. In this step are evaluated different video sequences from those used in the training phase.



QP	Width	...	Split
22	32	...	1
27	16	...	1
...	...	...	...
37	64	...	0

Figure 27 – Framework for training CU partitioning decision with LGBM models and evaluating the performance in the VTM encoder.

Table 7 presents the eight video sequences used in the training process with resolutions ranging from 416×240 up to 3840×2160 pixels (SHARMAN; SUEHRING, 2017; DAEDE; NORKIN; BRAILOVSKIY, 2019; XIPH, 2021).

Table 7 – Video sequences used for training.

Training Sequence	Resolution	Bit depth	FPS
TrafficFlow	3840×2160	10	30
BuildingHall2	3840×2160	10	50
Kimono1	1920×1080	8	24
ParkScene	1920×1080	8	24
Vidyo1	1280×720	8	60
Netflix_DrivingPOV	1280×720	8	60
pedetrian_area	832×480 (downsampled)	8	25
Flowervase	416×240	8	30

The video sequences used in the training process encompass a wide range of video characteristics (e.g., 8- to 10-bit depth and 24 to 60 frames per second – fps) for rendering several examples of block partitioning decisions in the training process.

The video sequences were encoded following the encoder configurations specified in JVET CTC for all-intra configuration, using QP values 22, 27, 32, and 37. We extracted the datasets based on 120 frames to reduce the training process complexity; these datasets were balanced according to the number of instances for each frame, block size, QP value, and output class.

7.3. Features Analysis and Selection

We collected a large amount of data from the video sequences and internal encoding variables to find features that could lead to effective decisions of CU split type. All these features were extracted directly during the encoding, where additional functions were implemented in the VTM encoder. These features encompass four information categories: CU samples, local samples, context, and coding information.

CU samples information considers features related to the current CU samples. All these features are computed based on luminance samples inside the whole CU, including width and height of the current CU, area, block ratio, variance (*var*), horizontal (*Gx*) and vertical (*Gy*) gradients based on Sobel operator, *Gx* divided by *Gy* (*ratioGxGy*), and the sum of *Gx* and *Gy* divided by the block area (*normGradient*).

The information of local samples refers to features obtained in smaller regions of the current CU, such as the absolute difference of variances on four sub-quarters (*diffVarQT*), maximum variance on four sub-quarters (*maxVarQT*), the absolute difference between variances of upper and lower regions of the CU (*diffVarHor*), and the absolute difference between left and right regions of the CU (*diffVarVer*).

Context information includes features of left, above, above-left, and above-right neighboring CUs, such as average QT (*neighAvgQT*) and MTT (*neighAvgMTT*) depth levels in neighboring CUs and number of neighboring CUs with QT (*neighHigherQT*) and MTT (*neighHigherMTT*) depth levels higher than the current CU.

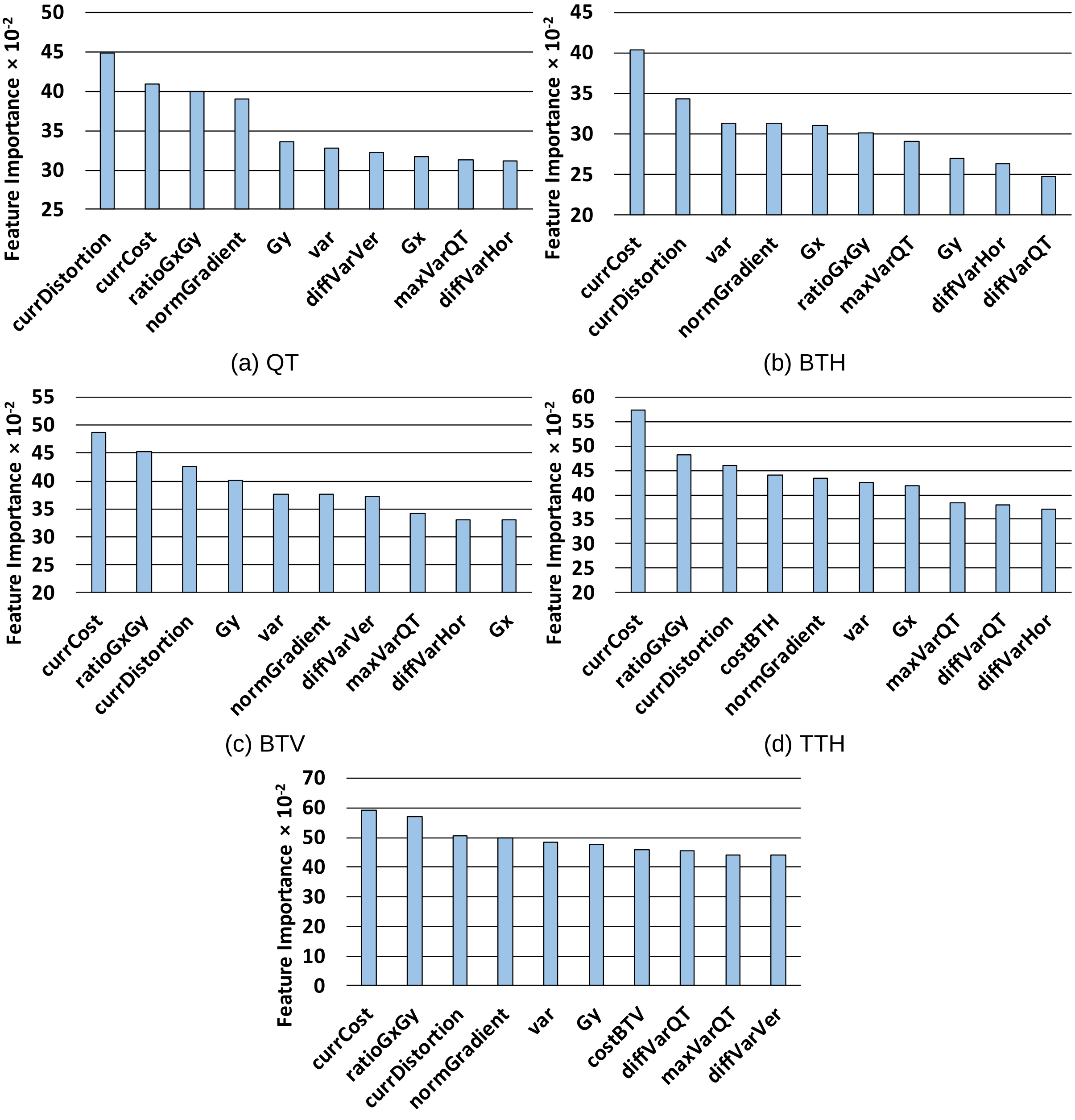
Since not split is evaluated first than QT, BT, and TT splits, we can consider several coding attributes obtained with the current CU size for deciding the split types. *Coding information* comprises coding attributes related to the current CU evaluated with not split type, such as QP, RD cost (*currCost*), distortion (*currDistortion*), current QT (*QTD*), BT (*BTD*), MTT (*MTTD*), and QTMT (*QTMTD*) depth levels, best intra prediction mode (*currIntraMode*), MRL index (*mrlIdx*), LFNST index (*lfnstIdx*), ISP mode (*ispMode*), and MTS flag (*mtsFlag*). Besides, since the split types are performed in order, the next split can take advantage of information obtained in the split evaluated previously, then the coding information also considers BTH RD cost (*costBTH*), BTV RD cost (*costBTV*), *costBTH* divided by *costBTV* (*ratioCostBTHBTV*), and TTH RD cost (*costTTH*). It is also important to mention that the corresponding RD-cost is unavailable when a previous split evaluation type is skipped, and, in this case, the feature is assigned with the maximum finite double-precision value.

Table 8 shows the features used in the QT, BTH, BTV, TTH, and TTV classifiers (19, 28, 29, 28, and 29, respectively), selected using the Feature Selector tool (KOEHRSEN, 2018). We removed collinear and low-importance features to reduce the dataset dimensionality and the computational cost of the training process.

Figure 28 presents the feature importance of the top 10 features for each classifier. The feature importance was measured using the split metric, which calculates the number of times the feature is used in the model. One can notice that features related to RD cost (*currCost* and *currDistortion*) followed by texture information have great importance for all classifiers. Besides, RD cost of previous split types also provides valuable information for the next split evaluations. Features indicating a horizontal texture direction such as *Gx* and *diffVarHor* are most important for horizontal splits (BTH (Figure 28(b)) and TTH (Figure 28(d)). While features indicating a vertical texture direction such as *Gy* and *diffVarVer* are most important for vertical splits (BTV (Figure 28(c)) and TTV (Figure 28(e)).

Table 8 – Features used for each classifier.

Feature	**Description**	**QT**	**BTH**	**BTV**	**TTH**	**TTV**
QP	The current QP value	×	×	×	×	×
currCost	The current RD-cost	×	×	×	×	×
currDistortion	The current distortion	×	×	×	×	×
width	The current block width	×	×	×	×	×
height	The current block height		×	×	×	×
area	The current block area		×	×	×	×
blockRatio	Block width divided by block height		×	×	×	×
QTD	The current QT depth level			×		
BTD	The current BT depth level		×	×	×	×
MTTD	The current MTT depth level		×	×	×	×
QTMTD	The current QTMT depth level		×	×		
currIntraMode	The current intra prediction mode	×	×	×	×	×
mrlIdx	Reference line index of MRL		×			
ispMode	Identify the ISP mode	×	×	×	×	×
mtsFlag	Identify the use of MTS		×	×	×	×
lfnstIdx	Identify the index of LFNST		×	×		
var	Block variance	×	×	×	×	×
diffVarQT	Absolute difference of variances on four sub-quarters	×	×	×	×	×
maxVarQT	Maximum variance on four sub-quarters	×	×	×	×	×
diffVarHor	Absolute difference between variances of upper and lower regions of the CU	×	×	×	×	×
diffVarVer	Absolute difference between variances of left and right regions of the CU	×	×	×	×	×
Gx	Horizontal Sobel gradient	×	×	×	×	×
Gy	Vertical Sobel gradient	×	×	×	×	×
ratioGxGy	Gx divided by Gy	×	×	×	×	×
normGradient	Sum of Gx and Gy divided by block the area	×	×	×	×	×
neighAvgQT	Average QT depth level in neighboring CUs	×	×	×	×	×
neighHigherQT	Higher QT depth level in neighboring CUs	×	×	×	×	×
neighAvgMTT	Average MTT depth level in neighboring CUs	×	×	×	×	×
neighHigherMTT	Higher MTT depth level in neighboring CUs	×	×	×	×	×
costBTH	RD-cost of BTH split type			×	×	×
costBTV	RD-cost of BTV split type				×	×
ratioCostBTHBTV	RD-cost of BTH divided by RD-cost of BTV				×	×
costTTH	RD-cost of TTH split type					×
Number of Features		**19**	**28**	**29**	**28**	**29**



(a) QT

(b) BTH

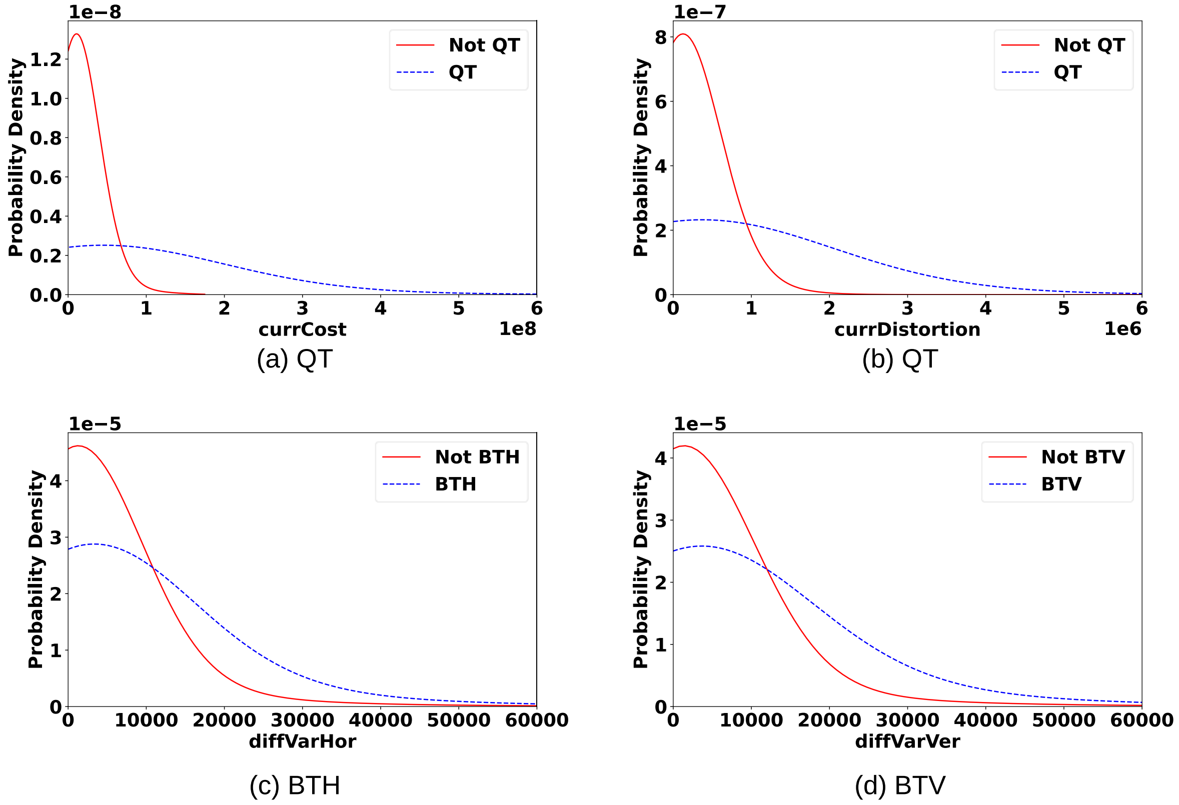
(c) BTV

(d) TTH

(e) TTV

Figure 28 – Feature importance ranking of top 10 features for (a) QT, (b) BTH, (c) BTV, (d) TTH, and (e) TTV classifiers.

Figure 29 exemplifies the probability density of four selected features for QT, BTH, and BTV classifiers. The attributes *currCost* and *currDistortion* demonstrate a clear correlation with the QT split, where low values of these attributes indicate a high probability of not splitting with QT. Also, low values of *diffVarHor* and *diffVarVer* indicate a high probability of not splitting with BTH and BTV, respectively.



(a) QT

(b) QT

(c) BTH

(d) BTV

Figure 29 – Probability density functions for (a) and (b) QT, (c) BTH, and (d) BTV classifiers regarding four analyzed attributes.

7.4. Classifiers Training and Performance

In the training process of classifiers, a crucial step to maximize the model performance is the hyperparameter optimization. The LGBM brings several hyperparameters to provide higher accuracy and deal with overfitting and underfitting that need to be properly optimized. For this purpose, the hyperparameters of each classifier were optimized using the efficient Optuna framework (AKIBA et al., 2019) and applying the Tree-structured Parzen Estimator (TPE) (BERGSTRA et al., 2011) approach.

The main optimized hyperparameters and the best values obtained for each classifier are presented in Table 9. *Learning_rate* corresponds to how quickly the error is corrected from each iteration (or tree) to the next. *Feature_fraction* specifies the percentage of features used for each iteration. *Bagging_fraction* specifies the fraction of data (training examples) used for each iteration, whereas *bagging_freq* indicates the frequency k for performing bagging. *Num_leaves* denotes the maximum number of leaves in one tree, and *Max_depth* limits the maximum depth for each tree. Finally, *Num_iterations* specifies the number of boosting iterations (or the number of trees).

Table 9 – Optimized hyperparameters for each classifier.

Hyperparameter	QT	BTH	BTV	TTH	TTV
learning_rate	0.10	0.13	0.12	0.12	0.14
feature_fraction	0.84	0.70	0.81	0.84	0.92
bagging_fraction	0.73	0.71	0.97	0.86	0.98
bagging_freq	3	5	7	7	1
num_leaves	254	250	231	251	253
max_depth	24	60	13	39	34
num_iterations	176	176	256	268	298

After the hyperparameter optimization process, the classifiers were evaluated using the 10-fold cross-validation, considering accuracy and F1-score metrics (APPENDIX B). The accuracy measures the ratio of correct predictions over the total number of instances evaluated and F1-score is the harmonic mean between precision and recall values (HOSSIN; SULAIMAN, 2015). Table 10 presents the accuracy, and F1-score results for each classifier, demonstrating that the classifiers obtain stable results for both metrics and can provide high performance to predict the CU split type.

Table 10 – Accuracy and F1-score results for each classifier.

Metric	QT	BTH	BTV	TTH	TTV
Accuracy	83.50%	74.69%	74.77%	76.52%	76.52%
F1-score	83.18%	74.64%	74.72%	76.59%	76.58%

The proposed solution follows the hierarchical process of VTM, and it uses the LGBM classifiers to avoid the evaluation of split types that have a low probability of being chosen as optimal partitioning. Since our solution encompasses five classifiers, each classifier indicates a probability value to skip the evaluation of a determined split type. By default, the decision threshold used by the LGBM model is 0.5, and the confidence of prediction is given by how close to 0 or 1 is the decision function output. If the output is higher than 0.5, the classifier decides to skip the split type evaluation; otherwise, the classifier remains the split type evaluation. However, the decision threshold can be configured, and different tradeoff results between encoding time reduction and encoding efficiency can be achieved.

Figure 30 displays the performance of each individual classifier implemented in VTM, regarding encoding time reduction and BDBR impact for the following thresholds: 0.3, 0.4, 0.5, 0.55, 0.6, 0.65, and 0.7. This evaluation allows the analysis of the individual results of each classifier for different operation points to validate their performance in terms of timesaving and BDBR. Note that the lower the threshold value, the higher the encoding time reduction and BDBR impact since more splits are skipped. In contrast, the higher the threshold value, the lower the encoding time reduction and BDBR impact since more splits are evaluated. Therefore, threshold values 0.3 and 0.7 provide the highest and the lowest time savings for all classifiers, respectively.

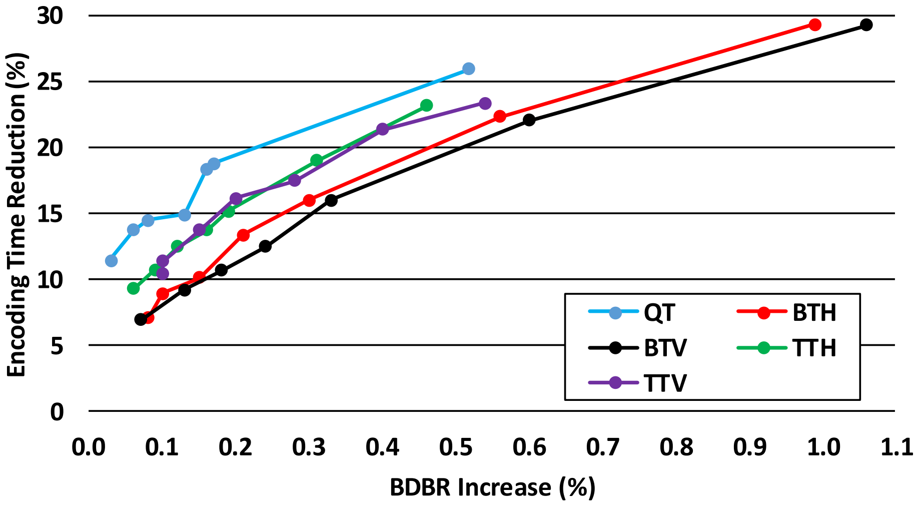


Figure 30 – Encoding time reduction and coding efficiency of each classifier for seven threshold values.

At this point, it is necessary to highlight that even though this first evaluation showed a low impact on the coding efficiency for each classifier regarding different threshold values, the integration of all classifiers is not a trivial task, and some adaptations were needed, as presented in the next section.

7.5. Classifiers Integration

Figure 31 presents the flowchart of the proposed solution composed of the five LGBM classifiers integrated with the QTMT splitting process. The white and light gray colors refer to native steps of the VTM encoding flow, and the green, blue, and orange colors represent the new steps of our solution introduced in the encoder, including feature extraction, classifier evaluation, and split evaluation decision, respectively.

After evaluating the intra-frame prediction with the not split type, our solution extracts the features to feed the LGBM classifiers. Subsequently, the VTM encoder verifies the split type that could be evaluated and according to this split type, an LGBM classifier is applied. Each LGBM classifier gives a probability to skip the evaluation of the corresponding split type. The probabilities obtained with the classifiers for QT, BTH, BTV, TTH, and TTV are P(QT), P(BTH), P(BTV), P(TTH), and P(TTV), respectively.

These probabilities are compared with decision thresholds to skip the evaluation of split types with a low probability of choosing as the best one. Then, the evaluation of a determined split type is skipped if the probability is higher than the decision threshold; otherwise, the encoding flow remains without modifications. The VTM split type evaluation is performed sequentially; thus, the proposed solution also performs a sequential decision, as shown in the flowchart. This approach takes advantage of the information of previously evaluated split types with specialized classifiers for each split type, intending to increase the accuracy of the decisions.

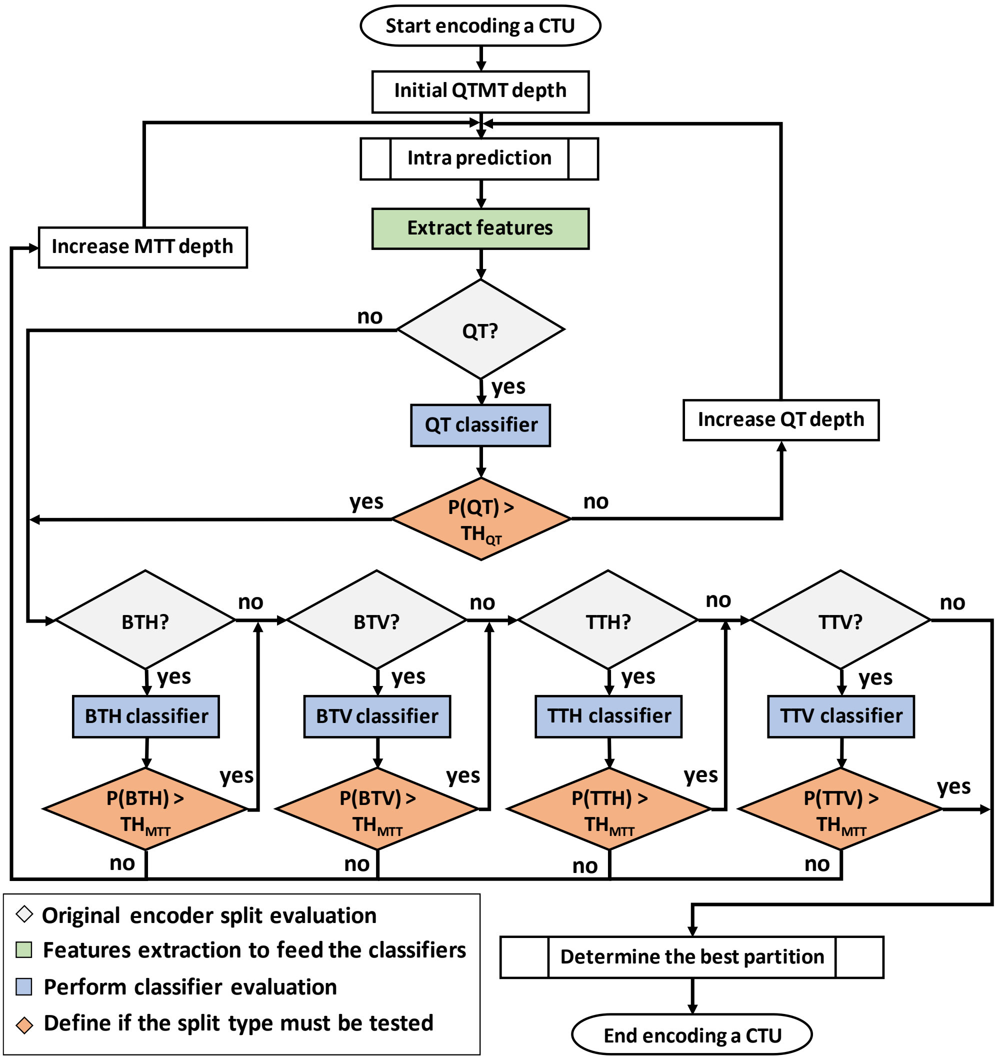


Figure 31 – Flowchart of the proposed solution integrated with the QTMT splitting process.

In the proposed solution, we established two decision thresholds to provide more flexibility: one for QT, called TH_{QT}, and another one for MTT (including horizontal and vertical BT/TT partitions), called TH_{MTT}. Our experimental analysis shows that using these two thresholds provides more flexibility for the proposed solution than using only one. In contrast, employing different decision thresholds for horizontal and vertical BT/TT partitions did not significantly modify the results.

Additionally, we have noticed that when our solution decides to skip all MTT splits in a given direction and the best partitioning would be in that direction, significant coding efficiency loss is caused. Therefore, the proposed solution skips all splits in a given direction (e.g., BTH and TTH) only if all splits have a high probability of being skipped (empirically defined as 0.7); otherwise, the split type with the lowest probability is evaluated. Considering these integration decisions, the QTMT split process defined in Figure 31 was implemented inside VTM to substitute the original VVC intra-frame prediction QTMT split process and the evaluations are presented in the next section.

The use of decision thresholds makes our solution highly configurable, providing multi-operation points and allowing the adaptation for different application requirements. Our solution defines the thresholds before encoding a video sequence according to the desired operating point. The change of the operation point can be done at multiple levels, according to the user's need, including CTU level, frame level, GOP level, or video level. The experiments presented in the next section consider the last option. Considering the flowchart presented in Figure 31, the change of operation point is done by changing the TH_{QT} and TH_{MTT} values.

Our solution accepts different combinations of thresholds to maximize the encoding timesaving or minimize the coding efficiency loss. On the one hand, increasing the threshold values reduces the number of split types skipped, allowing the encoder to evaluate more split types, and resulting in higher coding efficiency. On the other hand, decreasing the threshold values increases the number of split types skipped, resulting in a higher encoding time reduction.

In the next section, we present the evaluation of the proposed method using a configuration level with five operation points. These operation points were defined through an extensive experimental evaluation, showing good results to support different application requirements. However, the high flexibility of our solution enables even more combinations of threshold values to find the best operation point according to the application requirements. These five operation points are presented in Table V, considering the TH_{QT} and TH_{MTT}.

Figure 32 exemplifies the proposed solution in the QTMT split decision for a 32×32 CU, considering the C_3 operation point. Each classifier provides an output indicating the probability of skipping the associated split type. In the example of Figure

32, as the classifiers of QT, BTV, and TTH decided by skipping the evaluations, only BTH and TTV split types are evaluated for the current CU since only these splits have a lower probability than the decision threshold values.

Table 11 – Values of TH_{QT} and TH_{MTT} for the five operation points.

Configuration	TH_{QT}	TH_{MTT}
C_1	0.7	0.7
C_2	0.7	0.6
C_3	0.6	0.55
C_4	0.5	0.5
C_5	0.4	0.4

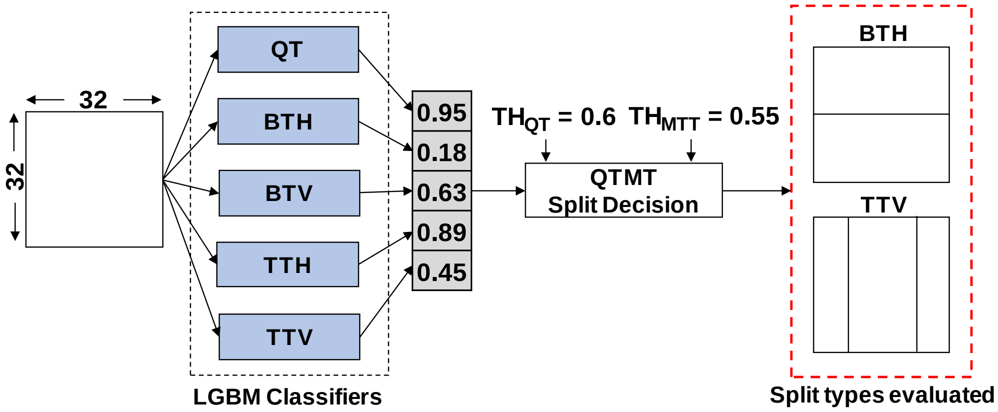


Figure 32 – QTMT split decision using the proposed solution for a 32×32 CU.

7.6. Results and Discussion

This section presents the results of the configurable fast-block partitioning decision solution for VVC intra coding using LGBM classifiers. Table 12 presents the encoding timesaving (ETS) and the coding efficiency, measured in BDBR, of our solution.

It is important to highlight that the training process did not use JVET CTC video sequences; thus, this evaluation considered different video sequences from the ones used in the training step, allowing a robust evaluation of the proposed solution.

These experiments considered the five operation points presented in the previous section and a configuration at the video level, meaning that the operation point does not change during the video sequence encoding. The results are presented for each video sequence, but the average and standard deviation are also presented to demonstrate the robustness of our solution considering different video characteristics and resolutions.

Table 12 – Encoding time saving and coding efficiency results of the proposed solution for five operation points.

Class	Video sequence	***Configuration Level***									
		C_1		*C_2*		*C_3*		*C_4*		*C_5*	
		BDBR (%)	**ETS (%)**	**BDBR (%)**	**ETS (%)**	**BDBR (%)**	**ETS (%)**	**BDBR (%)**	**ETS (%)**	**BDBR (%)**	**ETS (%)**
A1	Tango2	0.39	42.53	0.49	48.54	0.71	53.15	1.01	57.19	1.62	61.97
	FoodMarket4	0.35	36.12	0.50	41.28	0.69	46.55	0.96	51.18	1.51	56.82
	Campfire	0.39	29.40	0.59	38.85	0.83	43.76	1.18	51.82	2.02	59.57
A2	CatRobot	0.41	30.39	0.64	38.68	0.90	44.12	1.23	50.81	2.11	57.28
	DaylightRoad2	0.50	38.87	0.84	47.75	1.10	53.67	1.48	59.48	2.45	66.82
	ParkRunning3	0.21	28.70	0.31	35.58	0.45	41.36	0.61	47.60	0.98	52.12
B	MarketPlace	0.23	35.65	0.40	46.98	0.60	54.47	0.84	60.62	1.41	67.65
	RitualDance	0.52	40.04	0.79	46.23	1.09	53.46	1.53	58.88	2.54	65.40
	Cactus	0.45	34.70	0.73	44.42	1.04	49.99	1.47	57.24	2.58	64.76
	BasketballDrive	0.63	44.99	0.94	50.59	1.26	57.12	1.67	60.13	2.61	66.88
	BQTerrace	0.55	32.36	0.84	41.75	1.11	48.35	1.48	54.42	2.36	62.92
C	BasketballDrill	0.51	23.82	0.97	33.72	1.52	40.31	2.40	45.40	4.44	56.30
	BQMall	0.66	39.34	1.03	46.44	1.40	51.05	1.96	56.37	3.27	63.12
	PartyScene	0.27	35.90	0.52	41.74	0.77	48.33	1.15	53.66	2.11	61.16
	RaceHorsesC	0.34	32.63	0.53	41.36	0.75	46.95	1.07	52.53	1.97	61.01
D	BasketballPass	0.57	38.47	0.86	44.32	1.32	49.27	1.86	53.74	3.19	59.90
	BQSquare	0.21	26.51	0.39	35.61	0.57	40.66	0.85	47.05	1.58	56.36
	BlowingBubbles	0.27	29.87	0.52	38.14	0.82	43.71	1.19	48.98	2.27	57.15
	RaceHorses	0.28	28.52	0.45	37.35	0.72	45.04	1.06	49.90	2.08	56.86
E	FourPeople	0.83	40.33	1.24	47.99	1.71	54.11	2.38	59.09	4.04	66.54
	Johnny	0.97	43.92	1.27	49.98	1.65	55.30	2.13	59.37	3.45	65.65
	KristenAndSara	0.64	41.89	0.97	48.32	1.26	52.96	1.69	56.88	2.86	63.13
Average		**0.46**	**35.22**	**0.72**	**42.98**	**1.01**	**48.80**	**1.42**	**54.20**	**2.43**	**61.34**
Standard deviation (σ)		**0.20**	**6.00**	**0.28**	**5.12**	**0.36**	**5.12**	**0.50**	**4.65**	**0.85**	**4.36**

Table 12 displays that the five operation points (*C_1*, *C_2*,… ,*C_5*) provide an extensive range of ETS and BDBR values: from an ETS of 35.22% with a BDBR increase of 0.46% to an ETS of 61.34% with a BDBR increase of 2.43%.

The results showed that the proposed method can be efficiently applied to support various application requirements, with expressive ETS gains and minor impact in the BDBR results. These experiments also showed that our solution presented stable results for the evaluated video sequences, presenting low standard deviation for BDBR and ETS results, even considering different video characteristics and resolutions. Besides, the results outperform the state-of-the-art solutions in terms of combined rate-distortion and timesaving.

Figure 33 summarizes the comparisons with the related works. This figure presents a relation between ETS and BDBR for the related works and the five operation points previously defined in our solution. Nine related works were compared with our results, including the works (FU et al., 2019a; CHEN et al., 2020a; LEI et al., 2019; FAN et al., 2020; YANG et al., 2020; TISSIER et al., 2020; LI et al., 2021a; ZHAO et al., 2020; LI et al., 2021b).

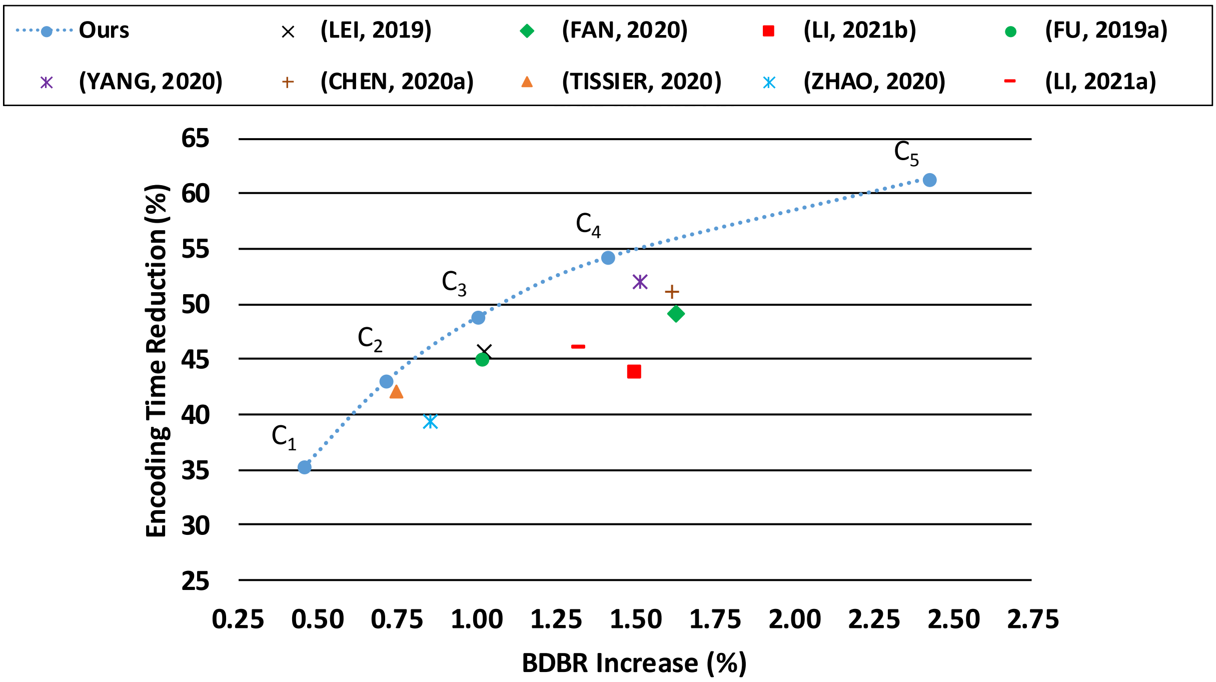


Figure 33 – ETS and BDBR increase for the five operation points of the proposed configurable solution and comparison with the related works.

The five operation points of our configurable solution are identified in Figure 33 as *C_1 to C_5*. This figure also presents a dotted line showing an extrapolation of our results if using other operation points with different threshold values.

Figure 33 clearly shows that our solution surpasses all related works since the results of the proposed solution achieved a better tradeoff between ETS and BDBR. This figure also clarifies the high level of flexibility provided by our configurable method compared to the related works since different relations between ETS and BDBR can be explored according to the application requirements.

Table 13 presents a more detailed comparison with some of these related works, where average BDBR and ETS results for each video class are presented, considering only *C_3* and *C_4* operation points for simplicity. These operation points were selected because they are the most comparable with the related works. The related works (FU et al., 2019a; CHEN et al., 2020a; YANG et al., 2020; LI et al., 2021a) were used in this comparison since they provide detailed results and used almost the same experimental setup considered in our work, making fairer this comparison.

Table 13 – Comparison with related works.

Class	**(FU, 2019a)**		**(YANG, 2020)**		**(CHEN, 2020a)**		**(LI, 2021a)**		**This work**			
									C_3		C_4	
	BDBR (%)	**ETS (%)**	**BDBR (%)**	**ETS (%)**	**BDBR (%)**	**ETS (%)**	**BDBR (%)**	**ETS (%)**	**BDBR (%)**	**ETS (%)**	**BDBR (%)**	**ETS (%)**
A1	1.31	51.00	0.85	51.39	1.17	42.71	1.60	43.90	0.74	47.82	1.05	53.39
A2	1.19	47.67	0.77	53.08	1.60	48.36	1.49	45.48	0.82	46.38	1.11	52.63
B	0.92	47.60	2.09	58.91	1.56	51.96	1.15	49.09	1.02	52.68	1.40	58.26
C	0.98	42.25	1.48	49.39	1.63	53.79	1.09	45.18	1.11	46.66	1.65	51.99
D	0.62	40.75	1.19	44.16	1.30	53.86	1.07	43.03	0.86	44.67	1.24	49.92
E	1.31	40.00	2.85	58.60	2.55	54.96	1.81	49.50	1.54	54.12	2.07	58.44
Avg	**1.02**	**45.00**	**1.52**	**52.01**	**1.62**	**51.23**	**1.32**	**46.13**	**1.01**	**48.80**	**1.42**	**54.20**
σ	**0.41**	**4.94**	**0.85**	**6.47**	**0.58**	**6.27**	**0.45**	**3.87**	**0.36**	**5.12**	**0.50**	**4.65**

When comparing our work with the solution of Fu et al. (2019a), one can observe that our operation point C_3 reached the highest ETS (48.80% compared to 45%) and a little bit smaller BDBR (1.01% against 1.02%). Our work reached similar standard deviation results for BDBR and ETS when compared with this solution.

Our operation point C_4, when compared with the work of Yang et al. (2020), reached a better ETS (54.20% against 52.01%) with a lower BDBR (1.42% against 1.52%). Our work also reached smaller standard deviation results for both ETS and BDBR when compared with this solution.

When compared with the work of Chen et al. (2020a), our solution at C_4 reached a better ETS (54.20% against 51.23%) with a lower BDBR (1.42% against 1.62%). Once more, our work reached a better standard deviation for both ETS and BDBR when compared with this solution.

Finally, when comparing our operation point C_3 with the solution of Li et al. (2021a), our work reached a better ETS (48.80% against 46.13%) with a lower BDBR (1.01% against 1.32%). When considering the standard deviation, our work reached a better result in BDBR, and a worst result in ETS when compared with this solution.

8. LEARNING-BASED FAST DECISION SCHEME FOR INTRA-FRAME PREDICTION MODE SELECTION OF LUMINANCE BLOCKS

The inclusion of new tools in the VVC intra-frame prediction significantly increased the number of candidate modes from the RD-list to be evaluated in the time-consuming RDO process, requiring solutions that expressively reduce time while maintaining coding efficiency. This chapter presents the learning-based fast decision scheme, which can discard intra prediction modes from the RD-list before the RDO process. This scheme is composed of three solutions: (i) a fast Planar/DC decision based on decision tree classifiers, (ii) a fast MIP decision based on decision tree classifiers, and (iii) a fast ISP decision based on the block variance. This solution was published in (SALDANHA et al., 2021c).

The analysis presented in Chapter 5 showed that the mode selection distribution between the conventional intra-frame prediction approach (AIP+MRL) and the new intra coding tools (MIP and ISP) can be explored. On average, AIP+MRL is selected 68% of the times, followed by MIP and ISP selected 24% and 8% of the times, respectively. Besides, Figure 34 demonstrates that Planar and DC modes are the most selected prediction modes in the conventional intra-frame prediction approach, occurring about 43% of the times. Among the angular modes, the most selected one (mode 50) occurs less than 5% of the times.

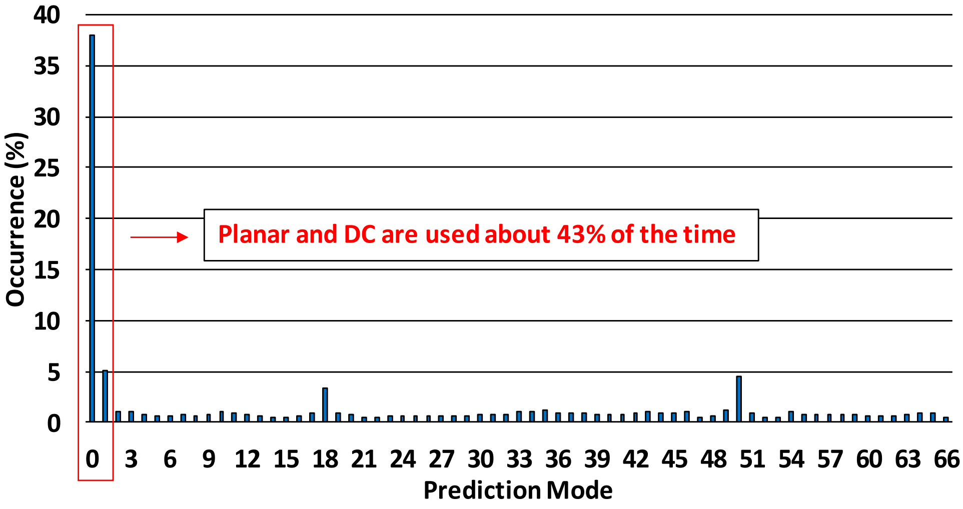


Figure 34 – Mode selection distribution for conventional intra-frame prediction approach.

These results allow us to conclude that, for most cases, the encoding selects the conventional intra-frame prediction approach. Besides, when the conventional intra-frame prediction is selected, Planar and DC modes are frequently used in the encoding process. Therefore, an intelligent encoding time reduction scheme to decide when to avoid the evaluation of angular, MIP, and/or ISP modes in the RDO process can provide high encoding timesaving with negligible impact on the coding efficiency.

In this solution, we also employed data mining to discover strong correlations between the encoding context and its attributes, but for defining decision tree classifiers that determine when discarding some prediction modes from the RD-list. For this purpose, we further implemented VTM functions and collected statistical information from the encoding process to offline train the decision tree classifiers using the REPTree algorithm in the *Waikato Environment for Knowledge Analysis* (WEKA) (HALL et al., 2009). The datasets were balanced according to the number of instances for each frame, block size, QP value, and output class. The training process employed the same video sequences, QP values and software configurations detailed in Chapter 7. We also evaluated more complex classifiers for this solution (e.g., LGBM); however, since these classifiers have a higher inference time than single decision trees and our space for coding time reduction in this encoding module (mode selection of intra-frame prediction) is lower than in the block partitioning module, the obtained encoding time reduction with these complex classifiers was not interesting.

8.1. Fast Planar/DC Decision based on Decision Tree Classifier

Since the conventional intra-frame prediction selects Planar and DC for many cases, which are frequently used to encode smooth texture regions, our first solution identifies when the smooth modes (Planar and DC) tend to be selected as the best prediction modes. In this case, our solution removes the angular modes from the RD-list, avoiding their evaluations in the RDO. For this purpose, we collected a large amount of data from the encoding process and defined a decision tree classifier to decide when the RDO evaluation of the angular modes can be avoided since these modes are unlikely to be chosen as the best ones. Table 14 displays the features used in the decision tree classifier, the corresponding descriptions, and the Information Gain (IG) (COVER; THOMAS, 1991).

Table 14 – Features used in the Planar/DC classifier.

Feature	Description	IG
QP	The current QP value	0.094
area	The current block area	0.040
dcInRdList	Notify if DC is on RD-list	0.147
posPlanar	The position of the Planar in the RD-list	0.309
posDC	The position of the DC in the RD-list	0.270
dcIsMPM	Notify if DC is an MPM	0.130
smoothIsFirst	Notify if Planar or DC is on the first RD-list position	0.214
mipIsFirst	Notify if MIP mode is on the first RD-list position	0.031
numAngModes	Number of angular modes in the RD-list	0.350
numMipModes	Number of MIP modes in the RD-list	0.174
numModesList	Total number of modes in the RD-list	0.130
dcSATDCost	SATD-cost of DC	0.281
planarSATDCost	SATD-cost of Planar	0.192
firstSATDNoSmooth	SATD-cost of the first angular or MIP in the RD-list	0.005
ratioSATDSmooth	SATD-cost of Planar divided by SATD-cost of DC	0.094
ratioSATD	Minimum SATD-cost between Planar and DC divided by firstSATDNoSmooth	0.201

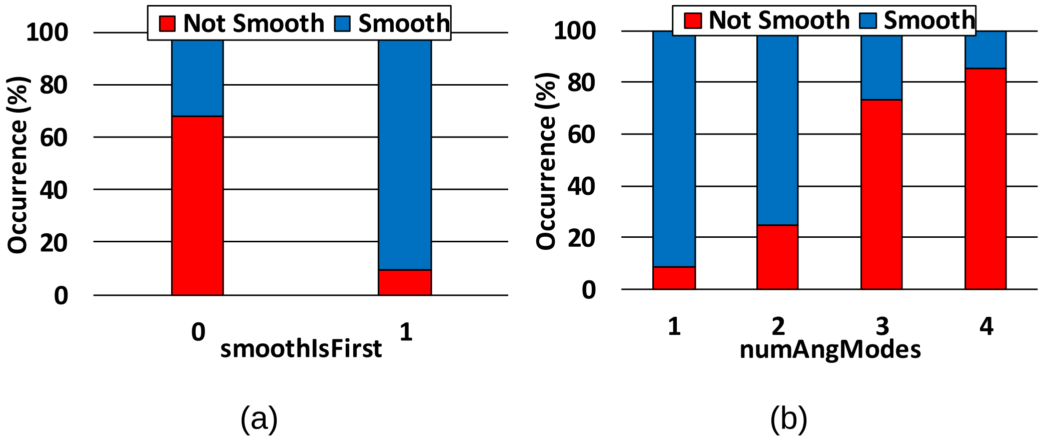


Figure 35 – Correlation between *smoothIsFirst* and *numAngModes* features with the smooth mode decision.

As examples, Figure 35(a) and (b) show the correlation of *smoothIsFirst* and *numAngModes* features with the smooth mode decision, respectively. *Not Smooth* indicates when an angular mode is selected as the best mode, and *Smooth* specifies when Planar or DC is selected as the best mode. When *smoothIsFirst* is equal to 1, a smooth mode is selected about 91% of the times. Also, the smaller the value of *numAngModes*, the higher is the probability of a smooth mode be selected, demonstrating a high correlation of these features with the smooth mode decision.

Planar/DC decision tree was designed with tree depth eight and evaluated using 10-fold cross-validation, attaining an accuracy result of 85.37% (F1-score 85.4%). The size of this decision tree is 85.

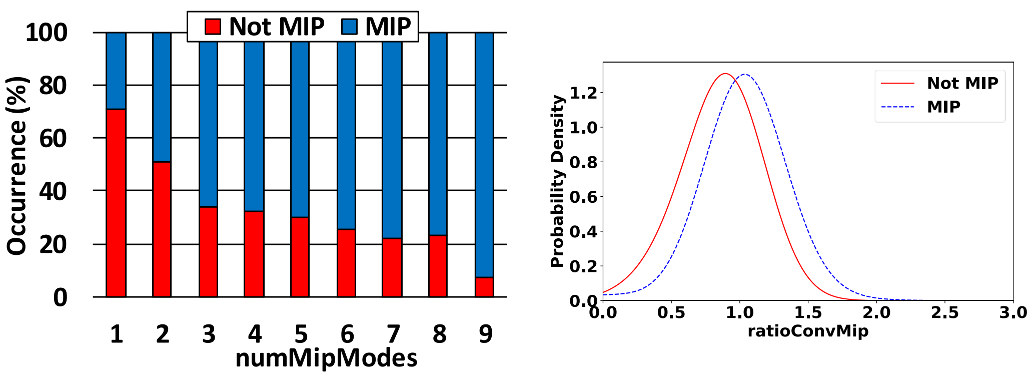
8.2. Fast MIP Decision based on Decision Tree Classifier

As demonstrated in Chapter 5, MIP modes are selected only about 24% of the times. Therefore, our second solution identifies when MIP modes can be discarded from the RD-list since they are unlikely to be selected in the RDO. This solution also collects a large amount of data from the encoding process to define a decision tree classifier for deciding when MIP modes can be removed from the RD-list, avoiding the RDO evaluation. Table 15 displays the features used in this decision tree classifier, the corresponding descriptions and IG.

Table 15 – Features used in the MIP decision tree classifier.

Feature	Description	IG
QP	The current QP value	0.093
width	The current block width	0.081
height	The current block height	0.080
area	The current block area	0.038
blockRatio	Block width divided by block height	0.055
QTMTD	The current QTMT depth level	0.109
posPlanar	The position of the Planar in the RD-list	0.138
posFirstMip	The position of the first MIP in the RD-list	0.295
numMipModes	Number of MIP modes in the RD-list	0.135
numModesList	Total number of modes in the RD-list	0.129
planarSATDCost	SATD-cost of Planar	0.005
firstSATDMip	SATD-cost of the first MIP in the RD-list	0.005
ratioConvMip	SATD-cost of the first conventional mode divided by the first MIP in the RD-list	0.230
numNeighMip	Number of neighboring blocks encoded with MIP	0.057

Figure 36(a) and (b) exemplify the correlation between *numMipModes* and *ratioConvMip* with the MIP decision, respectively. *Not MIP* and *MIP* indicate when a non-MIP or MIP mode is selected, respectively. The lower the value of *numMipModes*, the higher the probability of a non-MIP to be selected. Considering *ratioConvMip*, low values or values close to one tend to select a non-MIP mode, while values higher than one tend to choose a MIP mode. Both features demonstrate a high correlation with the MIP mode decision. MIP decision tree also was designed with tree depth eight and evaluated using 10-fold cross-validation, achieving accuracy results of 78.83% (F1-score 78.8%). The size of this decision tree is 97.

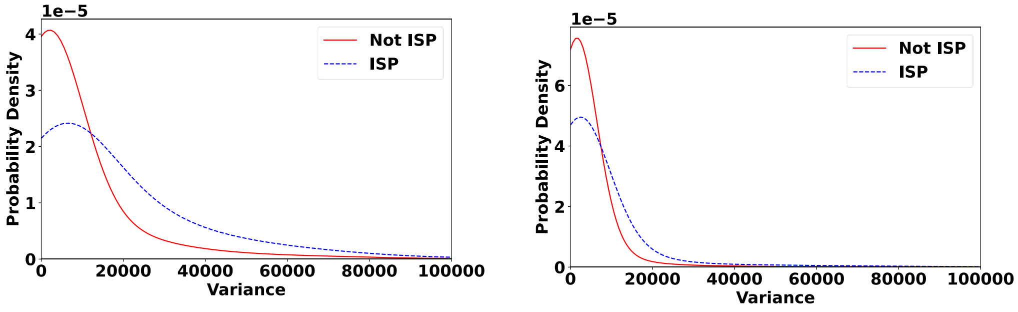


(a) (b)

Figure 36 – Correlation between two features and the MIP decision.

8.3. Fast ISP Decision based on the Block Variance

ISP is frequently used to encode more complex texture regions where the conventional intra-frame prediction approach cannot provide high coding efficiency. However, the conventional prediction can provide better rate-distortion results for simpler texture regions, allowing to avoid ISP evaluation. Based on this fact, our third solution analyzes the block texture complexity to decide when to remove ISP modes from the RD-list. Figure 37 shows the probability density functions for the block variance when the current block is predicted with ISP or not, considering Vidyo1 and Kimono1 video sequences encoded with QP 32.



(a) Vidyo1 (b) Kimono1

Figure 37 – Probability density function for ISP mode selection considering the block variance of different video sequences.

One can notice that the block variance can provide a good indication of when applying the ISP prediction and a threshold decision can be defined to remove ISP modes from the RD-list. These figures also demonstrate that a static threshold definition may deal with inaccurate decisions for some cases since it tends to vary according to the video content and QP value. Therefore, our solution uses the block variance to decide when removing the ISP modes, but the threshold value is computed online during the encoding of the first frame of the video sequence. Our solution stores the variance values of all blocks of the first frame that did not select ISP as the best mode and computes the average variance value. This variance value is used in the subsequent frames as a threshold to remove the ISP evaluations in the RDO. Besides, this threshold value can be periodically adjusted according to the application requirements, considering the number of frames, change of video content or, target timesaving.

8.4. Designed Scheme

Figure 38 presents the flowchart of the designed scheme to reduce the encoding time of intra-frame prediction mode selection.

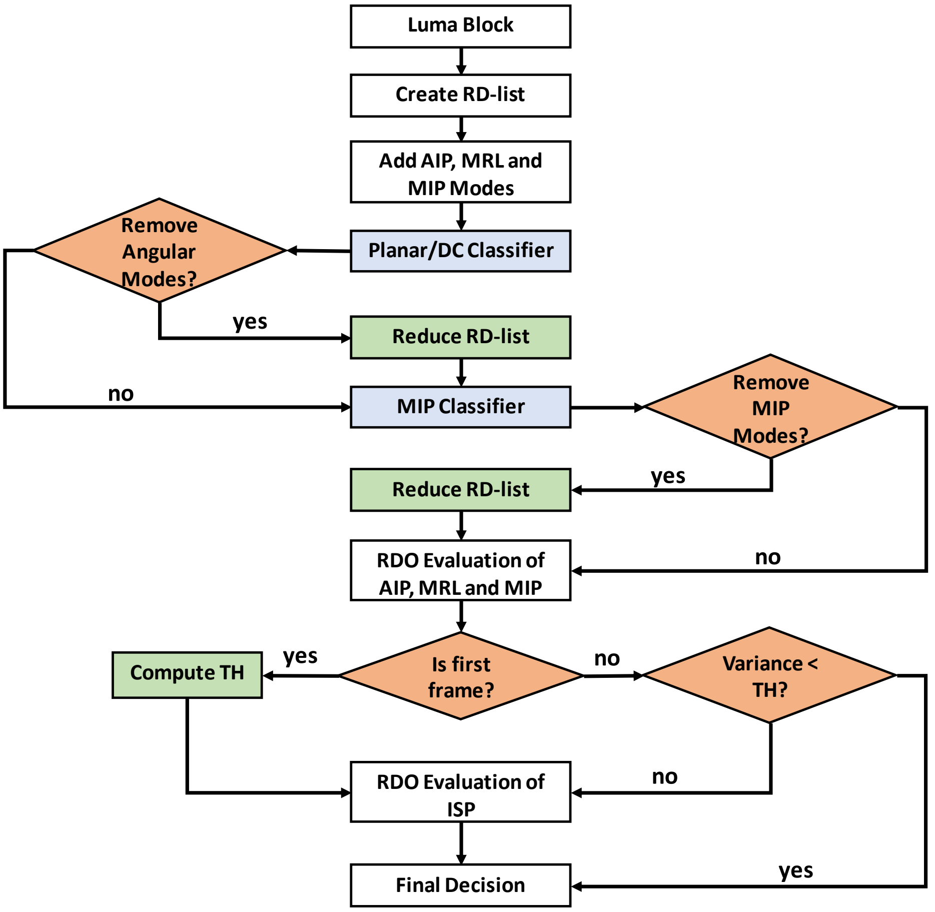


Figure 38 – Flowchart of the proposed scheme for VVC intra-frame prediction.

After inserting AIP, MRL, and MIP modes, the decision of the Planar/DC classifier is computed to verify if angular modes should be removed from RD-list. Similarly, the decision of the MIP classifier is computed to verify if MIP modes should be removed from RD-list, and RDO evaluates the resulting RD-list.

Subsequently, in the first frame of the video sequence, the ISP modes are evaluated by RDO and our technique performs the threshold calculation. For the remaining frames, our technique compares the block variance with the threshold value to verify if the ISP evaluation can be skipped.

8.5. Results and Discussion

This section presents the results of the proposed scheme for VVC intra coding. It is important to mention that VTM implements some native speedup heuristics for the intra-frame prediction process, such as a fast RMD search of the 65 angular modes, the evaluation of the MRL only for MPMs, and fast decisions for ISP based on the previous prediction mode evaluations as described in (DE-LUXÁN-HERNÁNDEZ et al., 2020). In our experiments, all these speedup techniques are enabled, allowing a fairer comparison with the current implementation of the VTM encoder.

Table 16 presents the ETS and BDBR results of Planar/DC and MIP DTs, fast ISP, and the overall proposed scheme. Planar/DC and MIP DTs reach 10.47% of ETS with a small increase of 0.29% in BDBR. Fast ISP reduces the coding complexity by 8.32%, increasing the BDBR by 0.31%. Combining all solutions, the proposed scheme can provide an ETS of 18.32% with a negligible BDBR increase of 0.60%. Besides, the proposed scheme presents a small standard deviation, demonstrating stable results for different video content and resolutions.

To the best of our knowledge, this is the first complexity reduction solution for VVC intra coding considering all the standardized intra-frame prediction tools; consequently, it is difficult to perform a fair comparison with related works. The works (YANG et al., 2020) and (CHEN et al., 2020b) used an old version of VTM (2.0) and reached 25.51% of ETS with a 0.54% BDBR increase and 30.59% of ETS with a 0.86% BDBR increase, respectively. Since our scheme targeted VTM with all standardized tools (e.g., MIP and ISP), having a more complex process to build the RD-list, one can conclude that our solution can provide high coding time savings with a minimum impact on the coding efficiency.

Table 16 – Proposed scheme results for CTC under all-intra configuration.

Class	**Video Sequence**	**Planar/DC and MIP Decision Trees**		**Fast ISP**		**Overall**	
		BDBR (%)	**ETS (%)**	**BDBR (%)**	**ETS (%)**	**BDBR (%)**	**ETS (%)**
A1	Tango2	0.50	11.90	0.08	7.92	0.56	18.89
	FoodMarket4	0.29	8.60	0.07	6.58	0.36	15.29
	Campfire	0.26	11.82	0.03	9.26	0.29	19.63
A2	CatRobot	0.33	9.94	0.24	7.01	0.58	17.62
	DaylightRoad2	0.39	12.80	0.24	9.23	0.64	19.71
	ParkRunning3	0.13	8.15	0.07	5.24	0.18	12.63
B	MarketPlace	0.11	12.47	0.08	10.67	0.22	20.84
	RitualDance	0.19	9.92	0.24	6.84	0.44	19.79
	Cactus	0.26	12.05	0.29	5.94	0.55	19.36
	BasketballDrive	0.42	10.20	0.43	7.95	0.84	19.36
	BQTerrace	0.36	10.29	0.37	10.22	0.72	20.07
C	BasketballDrill	0.41	6.25	0.69	7.99	1.11	16.05
	BQMall	0.33	10.31	0.66	9.43	0.94	19.40
	PartyScene	0.19	11.26	0.34	11.18	0.54	20.28
	RaceHorsesC	0.15	11.47	0.26	11.90	0.41	18.17
D	BasketballPass	0.36	9.98	0.41	7.46	0.72	16.08
	BQSquare	0.32	10.52	0.39	8.25	0.69	19.24
	BlowingBubbles	0.23	9.96	0.34	7.09	0.57	16.55
	RaceHorses	0.15	11.02	0.28	11.38	0.43	19.86
E	FourPeople	0.28	11.48	0.46	6.65	0.79	18.21
	Johnny	0.45	9.61	0.48	6.92	0.85	17.30
	KristenAndSara	0.38	10.32	0.43	7.83	0.83	18.75
Average		**0.29**	**10.47**	**0.31**	**8.32**	**0.60**	**18.32**
Standard Deviation (σ)		**0.11**	**1.50**	**0.18**	**1.85**	**0.24**	**1.98**

9. FAST DECISION SCHEME FOR INTRA-FRAME PREDICTION TRANSFORM SELECTION OF LUMINANCE BLOCKS

The insertion of new tools in the VVC transform coding significantly increased the number of encoding possibilities to be evaluated in the time-consuming RDO process, requiring solutions to reduce the computational effort while maintaining the coding efficiency. This chapter presents the fast transform decision scheme based on decision tree classifiers, which can avoid some evaluations of MTS and LFNST coding tools.

As demonstrated in Chapter 5, MTS transform matrices are less used than DCT-II. On average, MTS is selected 35% of the times, indicating that DCT-II or TSM are selected for most cases. The distribution changes a little bit according to the block sizes but with the small standard deviation. Moreover, on average, LFNST is used in 49% of the blocks, demonstrating that more than 50% of the cases are encoded without secondary transform. Besides, different block sizes show a higher variation in the results. From our experiments, the lowest and the highest percentage of LFNST use happen with 64×64 (33%) and 16×16 (66%) blocks sizes, respectively.

These results allow us to conclude that, in most cases, the blocks are encoded without the use of MTS or LFNST coding tools. Consequently, the evaluations of MTS and LFNST in the costly RDO process can be avoided for several cases. Therefore, an efficient encoding time reduction scheme able to decide when to avoid the evaluation of MTS and/or LFNST in the RDO can provide interesting timesaving with negligible impact on the coding efficiency.

For the development of this scheme, we employed the same methodology used in Chapter 8, but here the decision tree classifiers are responsible for determining when removing the evaluation of MTS and/or LFNST coding tools. In this case, single decision trees also presented better performance than more complex classifiers, as discussed in Chapter 8.

9.1. Fast MTS Decision based on Decision Tree Classifier

Since MTS is selected only about 35% of the times, and it is evaluated later than DCT-II and TSM in the VTM implementation, the coding information obtained prior to MTS evaluation can be used to detect when MTS evaluation is unnecessary. Based on the encoding context and the current coding information, our first solution identifies when the MTS evaluation can be skipped from the RDO process. We collected a large amount of data from the encoding process and defined a decision tree classifier for deciding when the MTS evaluation can be skipped. Table I demonstrates the features used in the decision tree classifier, the corresponding descriptions and IG.

Table 17 – Features used in the MTS decision tree classifier.

Feature	Description	IG
QP	The current QP value	0.016
area	The current block area	0.086
width	The current block width	0.116
BTD	The current binary tree depth	0.111
mipFlag	Notify if MIP was selected	0.009
ispMode	Identify the ISP mode	0.017
currCost	RD-cost	0.054
currDistortion	Total distortion	0.073
currFracBits	Number of encoded bits	0.090
currIntraMode	Intra prediction mode	0.068
noIspCost	RD-cost of the best non-ISP mode	0.117
numNonZeroCoeffs	Number of non-zero coefficients	0.026
absSumCoeffs	Absolute sum of the coefficients	0.094
numNeighMTS	Number of neighboring blocks encoded with MTS	0.030

Figure 39(a) and (b) exemplify the correlation between *numNeighMTS* and *ispMode* with the MTS decision, respectively. “Others” refers to the selection of DCT-II or TSM and “MTS” refers to the MTS selection. When the *numNeighMTS* value is zero, about 61% of the blocks are encoded without MTS. In contrast, the higher the value of *numNeighMTS*, the higher the probability of encoding a block with MTS. Considering *ispMode*, when this attribute has the value one or two, more than 62% of the blocks are encoded without MTS, whereas if the value is zero, 51% of the blocks are encoded with MTS. Both features demonstrate a correlation with the MTS decision.

MTS decision tree was designed with depth eight and size 125. This decision tree was evaluated using 10-fold cross-validation, achieving an accuracy result of 73.87% (F1-score 73.8%).

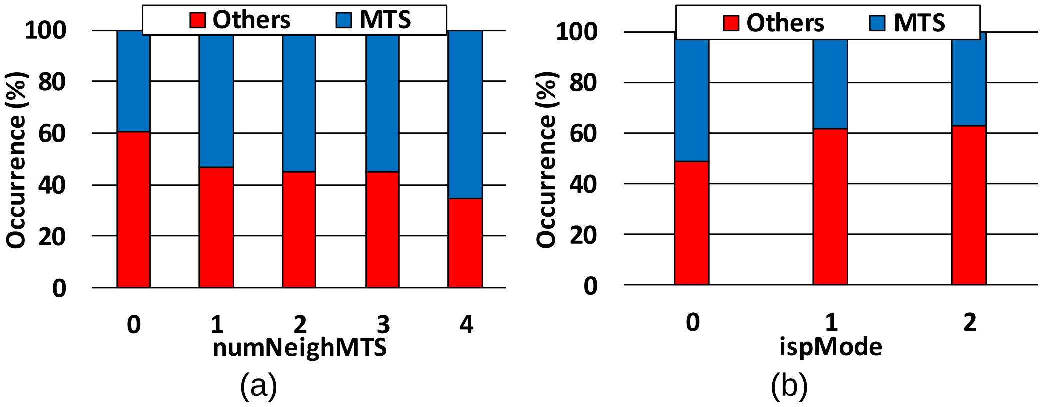


Figure 39 – Correlation between *numNeighMTS* and *ispMode* features with the MTS decision.

9.2. Fast LFNST Decision based on Decision Tree Classifier

Since from the blocks encoded with DCT-II, more than 50% are encoded without LFNST, our second solution identifies when the LFNST evaluation can be skipped. For this purpose, we collected a large amount of data from the encoding process when the DCT-II is evaluated without LFNST. Based on the collected data, we trained a decision tree classifier to decide when the LFNST evaluation can be avoided from the RDO process. Table 18 displays the features used in this decision tree classifier, the corresponding descriptions and IG.

Table 18 – Features used in the LFNST decision tree classifier.

Feature	Description	IG
width	The current block width	0.019
height	The current block height	0.170
area	The current block area	0.055
blockRatio	Block width divided by block height	0.021
MTTD	The current multi-type tree depth	0.176
mipFlag	Notify if MIP was selected	0.106
ispMode	Identify the ISP mode	0.015
currCost	RD-cost	0.086
currDistortion	Total distortion	0.136
currFracBits	Number of encoded bits	0.087
currIntraMode	Intra prediction mode	0.013
numNonZeroCoeffs	Number of non-zero coefficients	0.189
absSumCoeffs	Absolute sum of the coefficients	0.046

Figure 40(a) and (b) show the correlation between *MTTD* and *ispMode* attributes with the LFNST decision, respectively. When the *MTTD* value is zero, about 69% of the blocks are encoded without LFNST. In contrast, for higher values of *MTTD*, the percentage of blocks encoded with LFNST is about 52% (value 4). Regarding *ispMode*, this attribute presents a similar behavior when compared to the MTS decision tree classifier. When *ispMode* is one or two, more than 67% of the blocks are encoded without LFNST, and when this feature has the value zero, 50% of the blocks are encoded with LFNST, demonstrating a correlation with the LFNST decision.

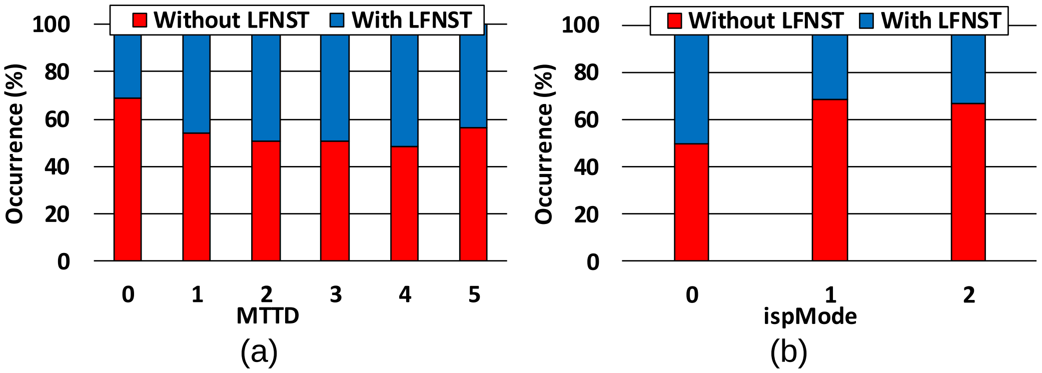


Figure 40 – Correlation between *MTTD* and *ispMode* features with LFNST decision.

LFNST decision tree was designed with depth eight and size 123. This decision tree was evaluated using 10-fold cross-validation, attaining an accuracy result of 76.75% (F1-score 76.7%).

9.3. Designed Scheme

Figure 41 presents the flowchart of the fast transform decision scheme using decision tree classifiers. After creating the RD-list, the encoder evaluates DCT-II (without LFNST) and TSM mode. Subsequently, based on the result of the transform coding with the lowest RD-cost, we compute the features for the MTS and LFNST decision tree classifiers and also compute the decision of the MTS and LFNST decision trees. When the MTS decision tree classifier decides to skip the MTS evaluation, the RDO computational effort is reduced; otherwise, no simplification is performed, and the MTS is evaluated. Analogously, when the LFNST decision tree classifier decides to skip the LFNST evaluation, the RDO process is simplified; otherwise, the LFNST coding evaluation remains without modifications.

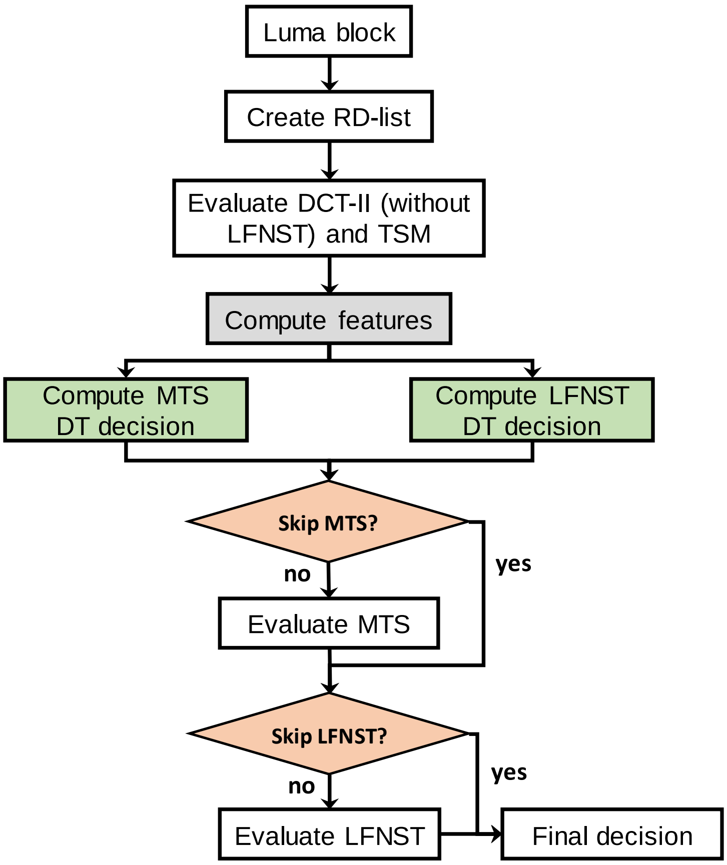


Figure 41 - Flowchart of the transform decision scheme.

9.4. Results and Discussion

This section presents the results obtained with the proposed fast transform decision scheme. Again, the training process did not use JVET CTC video sequences; consequently, this evaluation considered different video sequences from those used in the training step, allowing an unbiased evaluation of the proposed scheme. Besides, VTM implements some native speedup heuristics for the transform coding process, such as deciding to skip MTS and LFNST evaluations according to the ISP results and terminate the MTS evaluation according to the results of a certain MTS mode, as described in (ZHAO et al., 2021). In our experiments, all these speedup techniques are enabled, allowing a fairer comparison with the current implementation of the VTM encoder.

Table 19 presents the ETS and BDBR results of the MTS decision tree, LFNST decision tree, and the overall proposed scheme. MTS decision tree reaches 5% of ETS with a small increase of 0.21% in BDBR. The LFNST decision tree reduces the coding time by 6.40%, increasing the BDBR by 0.23%. The proposed scheme can provide an ETS of 10.99% with a negligible BDBR increase of 0.43% when combining both solutions. Besides, the proposed scheme presents a small standard deviation, demonstrating stable results for different video content and resolutions. The ETS and BDBR results range from 7.81% to 13.30% and from 0.15% to 0.70%, respectively.

Table 19 – Proposed scheme results for CTC under all-intra configuration.

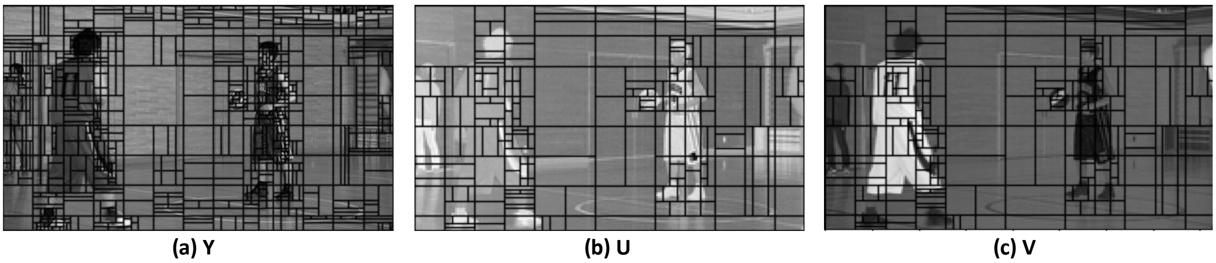
Class	Video Sequence	MTS Decision Tree		LFNST Decision Tree		Overall	
		BDBR (%)	ETS (%)	BDBR (%)	ETS (%)	BDBR (%)	ETS (%)
A1	Tango2	0.22	6.04	0.47	6.29	0.70	11.83
	FoodMarket4	0.27	2.72	0.41	5.26	0.68	8.91
	Campfire	0.17	3.25	0.16	4.21	0.32	8.24
A2	CatRobot	0.24	5.65	0.18	6.75	0.40	9.94
	DaylightRoad2	0.20	6.21	0.19	3.88	0.40	11.92
	ParkRunning3	0.21	2.76	0.12	6.67	0.30	7.81
B	MarketPlace	0.24	5.29	0.14	5.54	0.38	13.17
	RitualDance	0.27	5.37	0.32	7.62	0.60	7.96
	Cactus	0.23	3.89	0.19	3.90	0.40	11.52
	BasketballDrive	0.27	5.45	0.27	6.85	0.49	11.40
	BQTerrace	0.11	4.45	0.16	5.61	0.25	11.75
C	BasketballDrill	0.25	6.03	0.47	4.64	0.64	9.98
	BQMall	0.25	7.13	0.17	8.45	0.41	10.44
	PartyScene	0.10	5.93	0.08	9.58	0.20	12.70
	RaceHorsesC	0.17	4.81	0.18	7.10	0.32	13.14
D	BasketballPass	0.27	5.50	0.20	6.72	0.45	9.51
	BQSquare	0.09	4.39	0.07	6.68	0.15	13.30
	BlowingBubbles	0.14	5.20	0.12	7.81	0.26	12.20
	RaceHorses	0.17	4.91	0.17	8.64	0.34	11.74
E	FourPeople	0.33	4.29	0.36	7.51	0.66	11.23
	Johnny	0.28	4.67	0.33	4.60	0.58	10.49
	KristenAndSara	0.21	6.13	0.35	6.53	0.59	12.55
Average		**0.21**	**5.00**	**0.23**	**6.40**	**0.43**	**10.99**
Standard Deviation (σ)		**0.06**	**1.14**	**0.12**	**1.57**	**0.16**	**1.70**

To the best of our knowledge, this is the first encoding time reduction solution for VVC transform coding considering all the novel transform coding tools for VVC intra-frame prediction; consequently, it is difficult to perform a fair comparison with related works. The work of Fu et al. (2019b) used an old version of VTM (3.0) and reached 23% of ETS with a 0.16% BDBR increase; however, this solution also reduces the number of intra prediction modes in the RD-list and does not consider the LFNST coding tool. Since our scheme targeted VTM with all standardized tools (e.g., LFNST, MIP, and ISP), having a more complex process to decide the best encoding possibility, one can conclude that our solution can provide high coding time savings with a minimum impact on the coding efficiency. Besides, our scheme can also be combined with solutions to reduce the number of intra prediction modes in the RD-list or fast CU decisions to reach even more impressive timesaving results.

10. FAST BLOCK PARTITIONING SCHEME FOR CHROMINANCE BLOCKS

Although most of the coding effort is related to the luminance blocks, the encoding time of chrominance blocks cannot be neglected, especially targeting real-time video coding. For this purpose, we designed a fast block partitioning scheme for chrominance intra prediction, which explores the correlation between the chrominance and luminance coding tree structures and the statistical information of the chrominance samples in the current block. This solution was published in (SALDANHA et al., 2021d).

Figure 42(a), (b), and (c) present the block size distributions for luminance, chrominance (U), and chrominance (V), respectively, regarding the first frame of the BasketballPass video sequence, encoded with all-intra configuration and QP 37. Since VVC encodes chrominance channels U and V together, the same block size distribution is obtained for both components. One can notice that for most cases, chrominance is encoded with larger blocks than luminance since chrominance has more homogeneous regions. However, the chrominance QTMT structure evaluates all splitting possibilities to find the best one, demanding a high computational effort.


(a) Y

(b) U

(c) V

Figure 42 – Block size distributions for (a) luminance; (b) chrominance (U); and (c) chrominance (V).

For I-slices, luminance and chrominance QTMT structures are obtained separately for each CTU; however, the chrominance CTU encoding is performed after encoding the associated luminance CTU. As these channels represent the same scenario, one can explore some correlations between the luminance and chrominance QTMT structures. Furthermore, the characteristics of the chrominance CU samples can also be explored to identify the behavior of the texture and predict the best chrominance block partitioning type.

10.1. Chrominance CU Splitting Early Termination Based on Luminance QTMT

We propose the following idea to explore the correlation of the chrominance and luminance QTMT structures. For a given chrominance CU being evaluated in the

QTMT depth *d*, it is possible to verify the split type selected in the associated luminance CU for the same QTMT depth *d*.

Considering this fact, if the selected split type for the luminance CU in this depth is no split, there is a high probability that the best chrominance CU has been found at a depth lower or equal to *d*. Then, the process of dividing the chrominance CU ends; otherwise, the execution flow remains unchanged.

Figure 43 shows the success rate for five test video sequences (Flowervase, BasketballDrill, Netflix_DrivingPOV, ParkScene, and TrafficFlow) with resolutions ranging from 416×240 to 3840×2160, considering the four QP values defined in CTC. The success rate refers to the number of cases for which the proposed predictor had success divided by the total number of cases and it is calculated as follows:

$$Success\ Rate = \frac{Predictor\ have\ success}{Total\ cases}\ (\%) \tag{5}$$

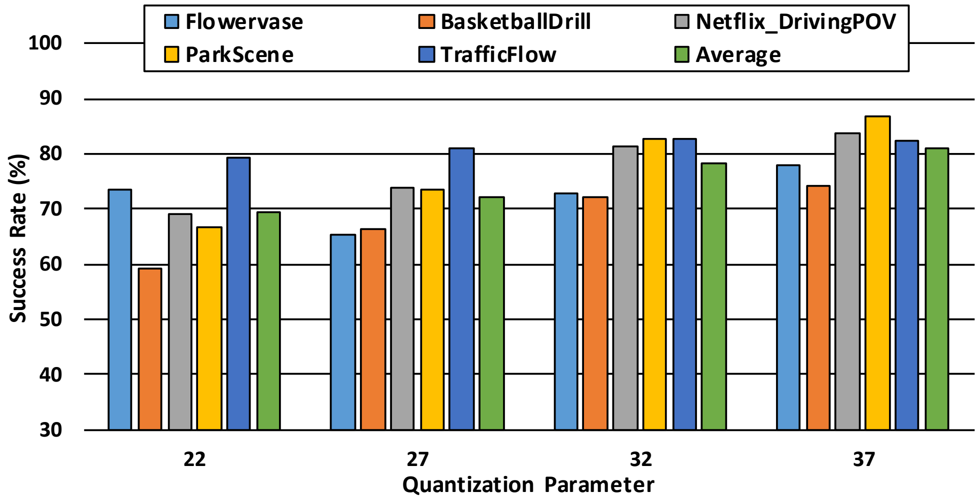


Figure 43 – Success rate of the best chrominance block size found in the QTMT depth lower or equal to luminance QTMT.

In this case, the total number of cases is the number of evaluated chrominance blocks. The predictor has success when the chrominance block is encoded with a QTMT depth lower or equal to the QTMT depth of the associated luminance block. On average, this predictor has a success rate higher than 70% for all QPs evaluated. Besides, the success rate also increases as QP increases, reaching up to 81% for QP 37. On average, a success rate of 75% was obtained, indicating that the luminance QTMT structure can be a good predictor for the chrominance QTMT coding.

Along with this analysis, Figure 44 presents the probability density functions for “not split” and “split” curves according to the RD-cost divided by the block area since larger blocks tend to have larger RD-costs. The block area refers to the number of samples inside the block, and it is calculated by multiplying block width by block height. The chrominance RD-cost was collected for both curves when the collocated luminance CU was defined as not split. The “not split” indicates that the chrominance CU was not split when the collocated luminance CU also was defined as not split. The “split” indicates that the chrominance CU was split when the collocated luminance CU was defined as not split.

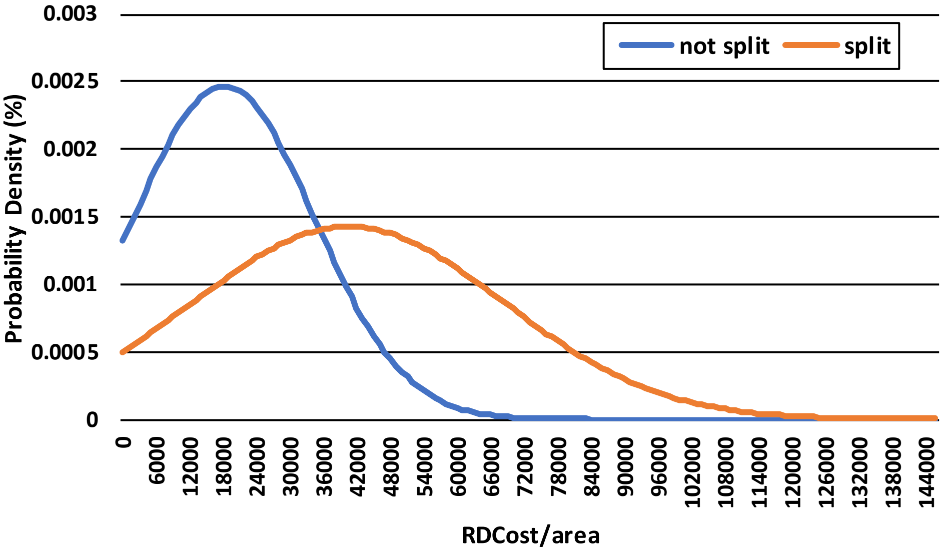


Figure 44 – Probability density function of splitting or not the chrominance CU using RD-cost based on the luminance QTMT.

These functions, which follow a gaussian distribution, were achieved by encoding the same video sequences of the previous analysis. These results demonstrate a high probability of the luminance QTMT predictor in having success for low RD-costs while having almost no chance to have success for larger values. Thus, this approach can be explored to perform an early termination decision according to a threshold criterion, increasing the success rate and decreasing the encoding efficiency loss.

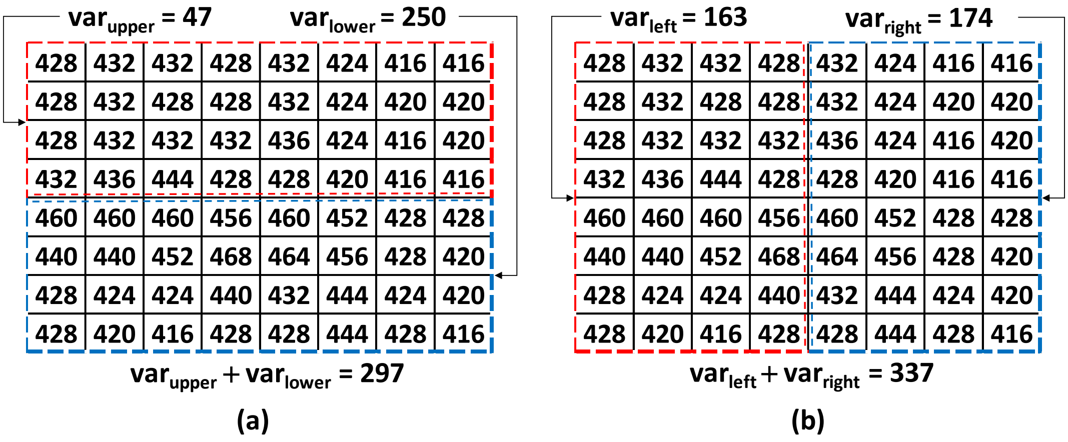
10.2. Fast Chrominance Split Decision Based on Variance of Sub-blocks

Typically, the best BT/TT partition direction is highly linked to the texture direction (FU et al., 2019a). This link happens because the best block partitioning is achieved with the partition type that results in a more homogeneous region or a region with similar behavior since it increases the accuracy of the prediction tools and reduces the residual error, incurring a smaller RD-cost.

Thus, we propose to evaluate the variance of sub-blocks in the current encoding chrominance CU as a predictor to decide the BT/TT partition directions. Figure 45 demonstrates the calculation of horizontal and vertical variances for an 8×8 chrominance CU in Figure 45(a) and Figure 45(b), respectively.

To calculate the horizontal variance, we consider that the current CU is horizontally subdivided into two equal-sized sub-blocks (similar to the BTH partitioning), resulting in two 8×4 sub-blocks (in the case of the example of Figure 45). The var_{upper} and var_{lower} are obtained by calculating the variances of the highlighted regions in red and blue, respectively. Then, var_{hor} refers to the sum of the variances of the upper (var_{upper}) and lower (var_{lower}) partitions. Similarly, var_{ver} considers that the current CU is vertically subdivided (similar to the BTV partitioning), where var_{left} and var_{right} are added to obtain the var_{ver}.

Considering that the lowest sum of variances indicates that the partition type provides more homogeneous regions, in this example of Figure 45, the vertical partitioning could be skipped since the horizontal partitioning is the most promising to achieve a better coding efficiency.



428	432	432	428	432	424	416	416
428	432	428	428	432	424	420	420
428	432	432	432	436	424	416	420
432	436	444	428	428	420	416	416
460	460	460	456	460	452	428	428
440	440	452	468	464	456	428	420
428	424	424	440	432	444	424	420
428	420	416	428	428	444	428	416

428	432	432	428	432	424	416	416
428	432	428	428	432	424	420	420
428	432	432	432	436	424	416	420
432	436	444	428	428	420	416	416
460	460	460	456	460	452	428	428
440	440	452	468	464	456	428	420
428	424	424	440	432	444	424	420
428	420	416	428	428	444	428	416

Figure 45 – Illustration of the calculation of (a) horizontal and (b) vertical variances for an 8×8 chrominance block.

The VVC encoder minimizes the RD-cost considering both chrominance components (U and V) to define the best block size and prediction mode combination. Consequently, the proposed predictor should also consider the features of both chrominance components to provide more accurate decisions. In this case, we calculate the var_{hor} and var_{ver} for the block samples of each chrominance component and only skip a given BT/TT direction if both components agree to skip that direction. For instance, the horizontal splitting is skipped only if var_{hor} (U) is lower than var_{ver} (U) and *var*$_{hor}$ (V) is lower than *var*$_{ver}$ (V). For simplifying, we use only *var*$_{hor}$ and *var*$_{ver}$ to represent the condition of chrominance U and V.

Figure 46 presents the success rate of skipping the horizontal partitioning (BTH and TTH) when *var*$_{ver}$ is lower than *var*$_{hor}$ and skipping the vertical partitioning (BTV and TTV) when *var*$_{hor}$ is lower than *var*$_{ver}$.

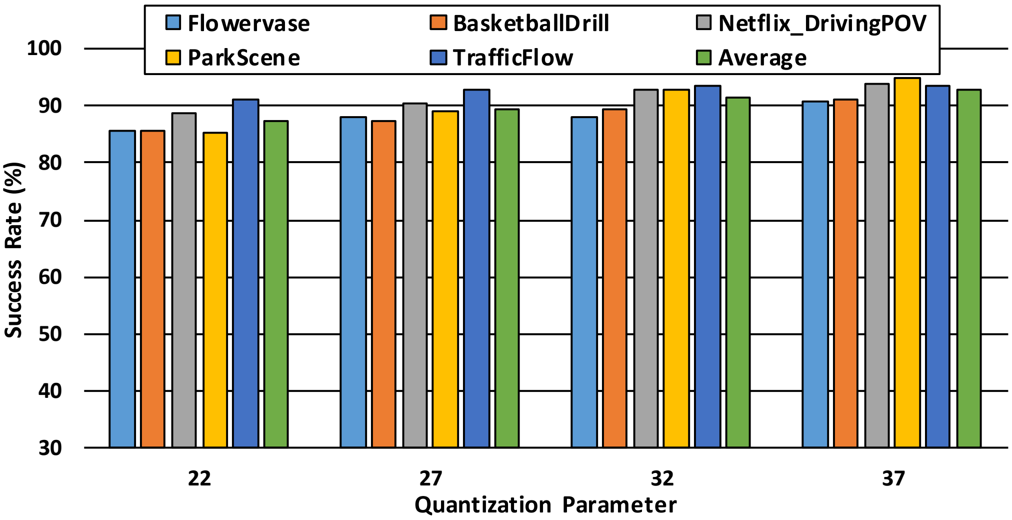


Figure 46 – Success rate of skipping the horizontal or vertical splitting of BT/TT partitions using the variance of sub-blocks.

Considering Equation (5), the predictor has success when it decides to skip the horizontal direction and the best split type selected is neither BTH nor TTH, or if the predictor decides to skip the vertical direction and the best split type selected is neither BTV nor TTV. The total number of cases is the total number of chrominance blocks that evaluated at least one horizontal split and one vertical split. This analysis also uses the same video sequences presented in Section 10.1 This evaluation shows that this predictor provides more than 85% of success rate for all evaluated cases. On average, this predictor presents a success rate of 90%, demonstrating that this solution can provide interesting results regarding encoding efficiency and timesavings.

10.3. Designed Scheme

Figure 47 presents the flowchart of the proposed fast block partitioning scheme, encompassing the Chrominance CU Splitting Early Termination based on Luminance QTMT (CSETL) and the Fast Chrominance Split Decision based on Variance of Sub-blocks (FCSDV).

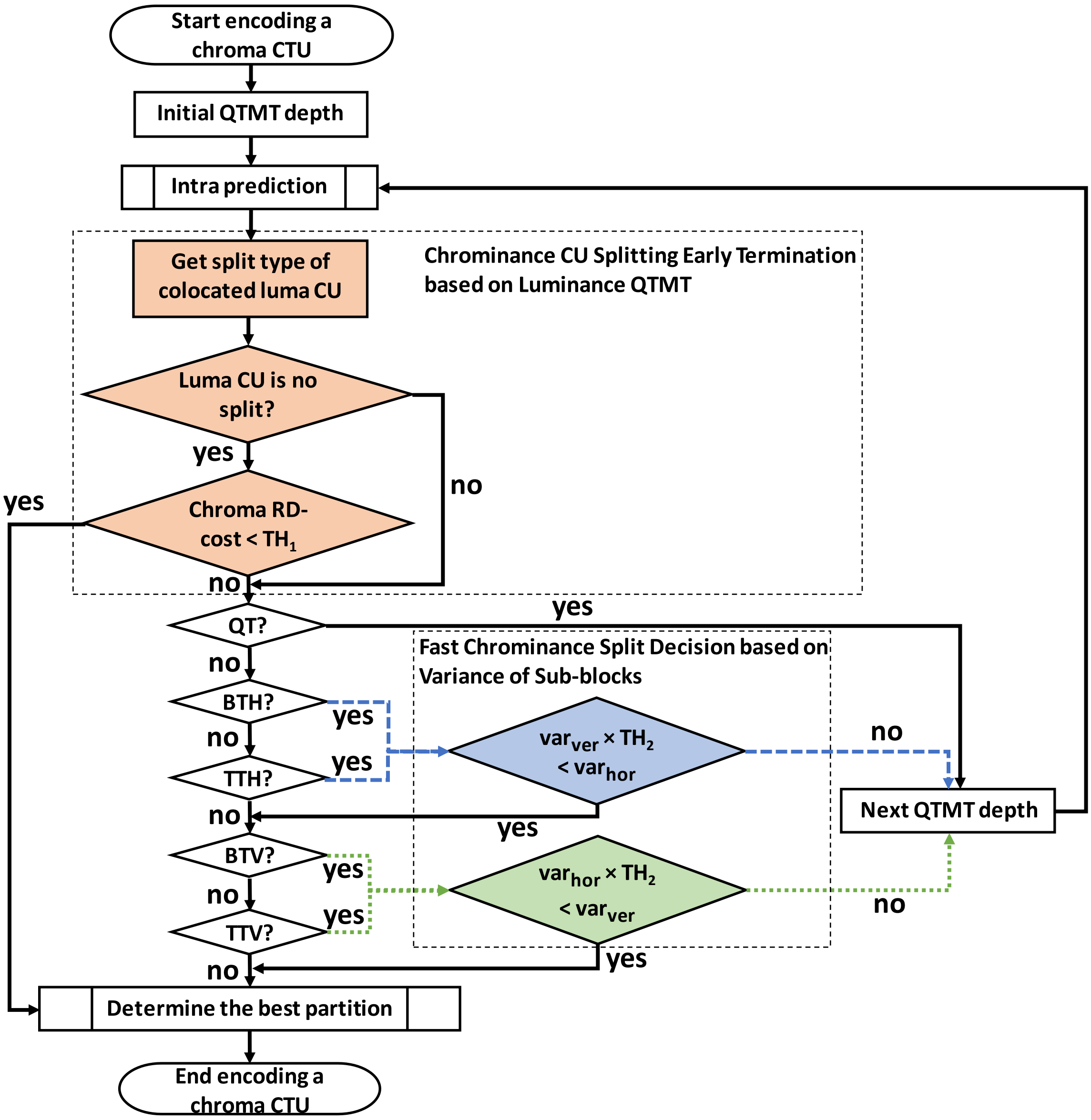

Start encoding a chroma CTU
Initial QTMT depth
Intra prediction
Get split type of colocated luma CU
Chrominance CU Splitting Early Termination based on Luminance QTMT
Luma CU is no split?
yes
no
Chroma RD-cost < TH₁
yes
no
QT?
yes
no
Fast Chrominance Split Decision based on Variance of Sub-blocks
BTH?
yes
no
TTH?
yes
var_ver × TH₂ < var_hor
no
yes
Next QTMT depth
no
BTV?
yes
no
TTV?
yes
var_hor × TH₂ < var_ver
no
yes
no
Determine the best partition
End encoding a chroma CTU

Figure 47 – Flowchart of the proposed fast block partitioning scheme for chrominance coding.

CSETL aims to terminate the chrominance QTMT evaluation early based on the luminance CU split type and the current chrominance RD-cost. When the luminance CU is "not split" and the chrominance RD-cost is lower than TH_1, the chrominance CU is not divided and the QTMT evaluation is finished; otherwise, the execution proceeds to evaluate other splitting possibilities.

FCSDV works to decide the direction of BT/TT partitions. On the one hand, when var_{ver} is lower than var_{hor} the horizontal partitions are skipped; on the other hand, when var_{hor} is lower than var_{ver} the vertical partitions are skipped. However, we decided to introduce another threshold criterion TH_2, because very close variance values have no obvious texture direction, hampering the definition of the best split direction. Therefore, TH_2 is a value that defines how many percent a variance value of a given direction should be less than the variance value of the other direction to avoid the evaluation.

We employed a detailed threshold analysis (using the video sequences presented in Section 10.1) that evaluates five scenarios for each solution of the proposed scheme, considering chrominance encoding time saving (C-ETS) and BDBR since the threshold values impact the encoding timesaving meaningfully.

BDBR can be calculated individually for each channel (Y, U, and V) or combined for the three components. For all cases, the overall bit rate is used to compute BDBR (i.e., the bits considering the three components). In contrast, different values of Peak Signal to Noise Ratio (PSNR) are used to compute the luminance (Y), chrominance (U), and chrominance (V) BDBR results, according to the corresponding component. YUV-BDBR also uses the overall bit rate, but it computes PSNR considering the three components through a weighted PSNR average (ITU-T, 2020), providing the coding efficiency result considering all components together. Since the proposed solution focuses on the chrominance coding and most of the bits are used to encode the luminance component, the individual chrominance BDBR can become difficult to interpret (ITU-T, 2020). Consequently, in addition to the chrominance BDBR results (U-BDBR and V-BDBR), we used YUV-BDBR to evaluate the quality impact in all video sequence channels and the total bit rate. It is important to highlight that all BDBR results of the previous sections considered only the BDBR (Y) since it is widely adopted in the literature for solutions focusing only on the luminance channel.

Figure 48 displays the C-ETS and YUV-BDBR results collected for different threshold values TH_1 used in CSETL. The average and the standard deviation of the RD-cost per area have been saved to define the evaluation scenarios. The evaluation scenarios for each threshold TH_1 use Equation (6), where k is empirically selected, ranging from 0 to 4, and μ and σ indicate the average and the standard deviation values for the TH computation, respectively.

$$TH_1 = \mu + k \times \sigma \tag{6}$$

CSETL allows various operation points for providing better coding efficiency or higher timesaving. We have separated two interesting limits to evaluate as a case study (Section 10.4). The TH1b in the solid red line provides a good tradeoff between YUV-BDBR and C-ETS, while the TH1d in the dotted blue line denotes the highest C-ETS.

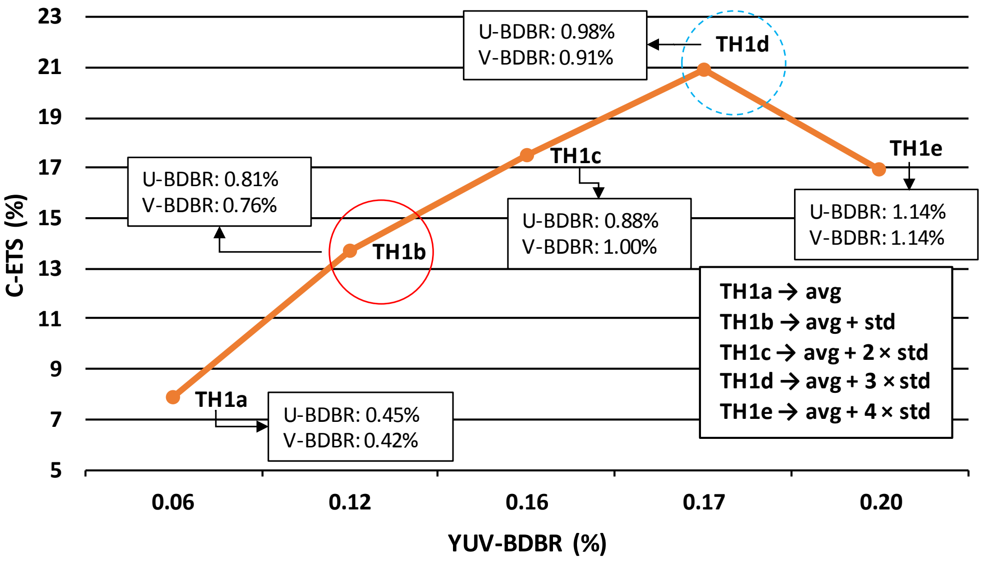


Figure 48 – Chrominance encoding timesaving and BDBR impact for CSETL solution according to some threshold values.

Figure 49 exhibits the C-ETS and YUV-BDBR results according to TH_2 for the FCSDV solution. Since the analysis in Section 10.2 presented a high success rate for this solution, we decided to verify the impact of using some threshold values and also without using a threshold ("noTH" in Figure 49). This threshold defines how many percent one variance must be less than the other for skipping BT/TT direction. For instance, TH_2=110% indicates that a given direction variance must be less than 10% of the other for skipping a direction.

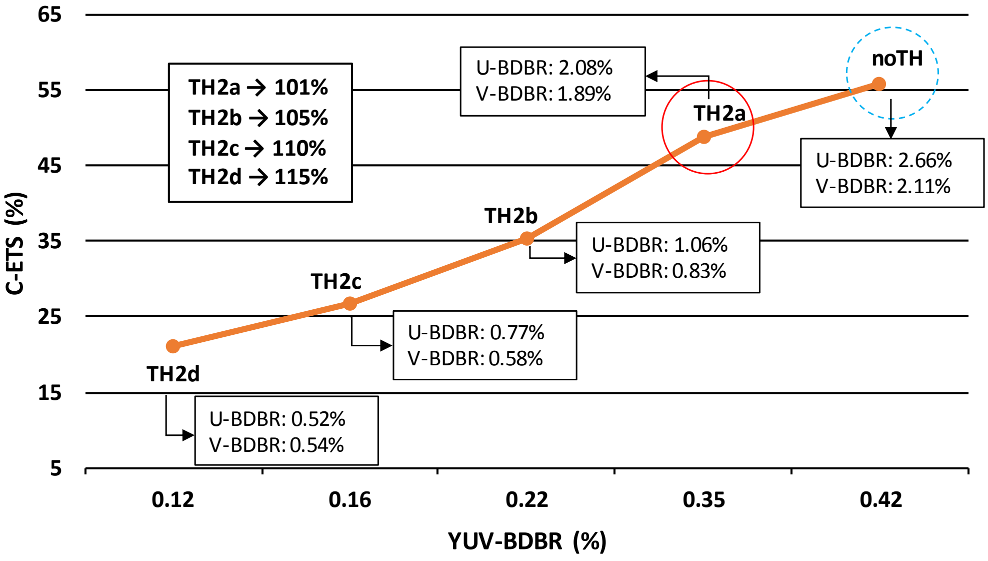


Figure 49 - Chrominance encoding timesaving and BDBR impact for FCSDV solution according to some threshold values.

In this case, the higher the threshold, the lower the C-EST and YUV-BDBR impact. Following the same idea as the previous experiment, we selected two case studies. We choose the highlighted threshold with a solid red line (TH2a) since it can avoid the cases where the variance values are very close, reducing the YUV-BDBR loss while maintaining a high C-ETS. The "noTH" evaluation provides the highest C-ETS reduction; this evaluation associated with TH1d of the CSETL solution can achieve the highest ETS of the proposed scheme.

Therefore, we selected two case studies considering two combinations of thresholds to be evaluated in the next section. **Case 1** refers to the combination of TH1b (CSETL) and TH2a (FCSDV), which provides the best tradeoff between coding efficiency and ETS. **Case 2** indicates the combination of TH1d (CSETL) and "noTH" (FCSDV), which allows the highest ETS of the proposed scheme.

10.4. Results and Discussion

This section presents the results of the fast block partitioning scheme for chrominance intra prediction, which encompasses CSETL and FCSDV solutions.

Table 20 shows the results acquired with the proposed solution for **Case 1** and **Case 2**. "Average (without TS *)" indicates the average of all video sequences excluding BasketballDrill that was used in the data collection step. C-ETS presents the time savings for chrominance only, and T-ETS presents the total time savings.

Case 1 is the configuration of thresholds that provides a good tradeoff between compression performance and C-ETS. In this case, the proposed scheme can save the chrominance and total encoding time by about 60.03% and 8.18%, with a negligible impact of 0.66% in YUV-BDBR, on average (without considering BasketballDrill). Negligible variation in the average results is obtained when the training sequence is considered, demonstrating that the proposed scheme can achieve excellent results regardless of the characteristics of the video sequence. On the one hand, the Campfire and ParkRunning3 video sequences produce the highest (72.97%) and the lowest (42.90%) C-ETS, respectively. On the other hand, Campfire and BQSquare result in the highest (1.79%) and lowest (0.20%) YUV-BDBR increase, respectively. These results demonstrated that the scheme could reduce more than 60% of the chrominance encoding time, on average, with minor impacts on the coding efficiency.

Furthermore, it is important to highlight that the highest C-ETS will not always result in the highest T-ETS since the luminance and chrominance encoding time distribution varies according to the video sequence and QP encoded. For instance, ParkRunning3 obtained 42.90% of C-ETS and 14.04% of T-ETS, whereas MarketPlace attained 60.49% of C-ETS and 6.97% of T-ETS.

Table 20 – Experimental results obtained with the proposed fast block partitioning scheme for chrominance coding under all-intra configuration.

Class	Video Sequence	Case 1					Case 2				
		YUV BDBR (%)	U BDBR (%)	V BDBR (%)	C-ETS (%)	T-ETS (%)	YUV BDBR (%)	U BDBR (%)	V BDBR (%)	C-ETS (%)	T-ETS (%)
A1	Tango2	1.07	7.91	8.40	64.29	4.87	1.11	8.26	8.69	65.75	5.72
	FoodMarket4	0.76	3.37	3.71	62.84	6.85	0.82	3.69	3.99	66.23	6.93
	Campfire	1.79	5.96	8.35	72.97	14.73	2.33	8.54	9.30	81.18	17.15
A2	CatRobot	1.54	7.70	7.32	68.73	11.76	1.93	9.51	9.12	74.77	13.16
	DaylightRoad2	0.39	4.98	3.74	58.52	4.77	0.46	6.07	4.32	64.08	6.56
	ParkRunning3	0.45	0.90	0.85	42.90	14.04	0.90	1.94	1.75	66.73	21.59
B	MarketPlace	0.55	3.90	2.95	60.49	6.97	0.72	5.15	3.56	68.86	8.95
	RitualDance	0.62	3.53	3.91	59.37	5.79	0.70	3.98	4.50	64.29	5.88
	Cactus	0.55	2.97	3.67	58.03	5.59	0.70	3.76	4.67	65.02	5.07
	BasketballDrive	0.64	3.17	3.70	58.42	4.11	0.75	3.78	4.35	63.71	6.23
	BQTerrace	0.21	3.12	2.92	53.95	6.24	0.28	3.72	3.54	61.61	7.09
C	BasketballDrill *	1.06	3.91	4.18	59.76	9.63	1.29	4.70	5.01	66.07	10.83
	BQMall	0.62	3.34	3.46	63.08	9.64	0.71	3.68	4.11	67.11	9.78
	PartyScene	0.39	2.35	2.53	57.50	10.02	0.50	2.95	3.26	63.94	11.59
	RaceHorsesC	0.52	1.61	2.43	64.20	10.78	0.67	1.81	2.97	70.57	12.28
D	BasketballPass	0.88	3.47	3.08	59.75	8.05	1.08	4.02	3.76	65.91	9.60
	BQSquare	0.20	1.67	2.42	53.18	5.16	0.22	1.76	2.69	58.80	5.77
	BlowingBubbles	0.42	1.80	2.33	57.61	9.74	0.49	2.48	2.61	62.86	9.18
	RaceHorses	0.71	2.18	3.04	62.47	10.28	0.85	2.56	3.82	69.80	11.82
E	FourPeople	0.36	1.74	1.91	57.23	6.22	0.41	1.90	2.33	60.96	6.25
	Johnny	0.72	3.73	2.26	62.51	6.51	0.76	3.92	2.74	65.99	8.07
	KristenAndSara	0.52	2.46	2.21	62.65	9.64	0.58	2.70	2.59	65.82	8.79
Average (without TS *)		**0.66**	**3.42**	**3.58**	**60.03**	**8.18**	**0.81**	**4.10**	**4.22**	**66.38**	**9.40**
σ (without TS *)		**0.39**	**1.83**	**1.96**	**5.86**	**2.99**	**0.50**	**2.19**	**2.12**	**4.75**	**4.04**

Considering **Case 2** that allows our scheme to achieve a higher C-ETS than other threshold combinations, the obtained average C-ETS and T-ETS were 66.38% and 9.40%, respectively, with a minor impact of 0.81% in YUV-BDBR. In this case, a higher time saving was obtained at the cost of a YUV-BDBR increase, if compared to **Case 1**. Regarding only the chrominance channels, **Case 1** increases the U-BDBR and V-BDBR by 3.42% and 3.58%, whereas **Case 2** impacts 4.10% and 4.22% in U-BDBR and V-BDBR, respectively.

To the best of our knowledge, this is the first solution to reduce the encoding time of chrominance block partitioning in the VVC encoder, presenting the results of chrominance timesaving and coding efficiency.

11. CONCLUSIONS AND FUTURE WORK

This Thesis proposed several algorithms using different approaches for different encoding modules to reduce the computational effort of the VVC intra-frame prediction. This Thesis was motivated by the high computational cost of the VVC standard due to the insertion of new coding tools to improve the coding efficiency. The VVC standardization adopted and proposed several new tools to efficiently encode high video resolution, increasing the encoding effort significantly compared to previous coding standards. Hence, VVC introduced new challenges concerning efficient real-time video processing, requiring efficient solutions to reduce computational cost.

As the first contribution of this Thesis, we can mention the extensive performance analysis of VVC intra-frame prediction presented in Chapter 5. The results proved that VVC provides substantial coding efficiency gain over HEVC, mainly for higher resolutions. Moreover, the encoding time and usage distribution analysis allowed us to identify the most time-consuming steps of VVC intra-frame prediction, which was crucial in developing our encoding time reduction solutions.

The contributions of this Thesis regarding time savings algorithms are: (i) encoding time reduction of the block partitioning for luminance samples; (ii) encoding time reduction of intra-frame prediction mode selection for luminance samples; and (iii) encoding time reduction of intra-frame prediction transform selection for luminance samples; (iv) encoding time reduction of the block partitioning for chrominance samples.

The topic described in (i) encompasses a statistical-based solution and a configurable solution based on machine learning to reduce the computational effort of the block partitioning structure. The statistical-based solution used the variance of block samples and the current prediction mode to decide when skipping horizontal or vertical split types of MTT structure, providing 29% encoding time reduction with a 0.8% BDBR increase. The configurable solution based on machine learning uses an LGBM classifier for each split type, which can decide when a determined split type can be skipped or not. This solution provides high flexibility with different operation points, reducing the encoding time from 35.22% to 61.34% and impacting from 0.46% to 2.43%.

The intra-frame prediction mode and transform selection using decision tree classifiers were presented in the topic (ii) and topic (iii), respectively. The mode selection encompasses two solutions using decision trees and one solution using an online statistical decision. The first and second decision trees decide when removing angular modes and MIP modes from the RDO evaluation, respectively. The online decision computes the variance of block samples and decides when skipping the ISP in the RDO evaluation. This solution provides 18.32% encoding time saving with a 0.60% BDBR increase. The transform selection encompasses two solutions using decision trees, where the first one decides when skipping the MTS evaluation and the second one decides when skipping the LFNST evaluation. This solution reaches 11% of encoding timesaving at the cost of 0.43% in BDBR.

Topic (iv) refers to the fast block partitioning scheme for chrominance block samples, where the solution uses the luminance block partitioning information and the variance of chrominance block samples to skip some split evaluations. This is the first solution in the literature focusing on the block partitioning of chrominance, providing results of coding efficiency and timesaving in the chrominance channel. This solution reaches more than 60% of chrominance encoding timesaving and 8% of total encoding timesaving, increasing about 0.7% of YUV-BDBR.

The proposed solutions in this Thesis presented competitive results compared to the related works. It is important to highlight that combining all these solutions in a VVC encoder can furnish good timesaving results but without providing the full potential of each method since they were designed individually. In developing each solution, we have considered the baseline VTM reference software (i.e., without modifications) to extract the features for the machine learning methods and compute the data for statistical analysis and threshold evaluations. Based on this fact, if we directly combine these solutions in the encoder, the solutions should consider another coding context; otherwise, more wrong decisions can be made than expected, decreasing the coding efficiency. The ideal scenario for this case is to incrementally introduce the proposed solutions into the reference software and specialize them accordingly with the new encoder context to maintain the high performance of the proposed solutions with a good tradeoff between timesaving and coding efficiency.

Summarizing this Thesis, it was designed several encoding time reduction solutions applying heuristic and machine learning approaches. Our evaluations demonstrated that significant timesaving was obtained applying these approaches with minor impact on the video quality, proving its claim. However, there is a vast space that still can be explored for achieving higher timesaving in the VVC encoder.

11.1. Future Work

We can note several open research possibilities not covered by the literature yet in the VVC encoder. The first possibility is to combine the proposed solutions in a VVC encoder by specializing the solutions to a context where they work together. Moreover, although each solution proposed in this work provides significant timesaving results for intra-frame prediction, there are many points that can be considered for future research. For instance, the improvement of the methods that apply machine learning by investigating novel features that are more meaningful to the transform or mode selection and the development of new solutions to speed up the mode selection of chrominance coding.

Besides, there are a few works in the literature focusing on the VVC inter-frame prediction, which is a module with high computational complexity in modern video coders. Solutions for the block partitioning structure in the inter-frame prediction were not explored enough. VVC also introduces novel coding tools for inter-frame prediction, such as affine motion compensation, geometric partitioning mode beyond the conventional motion estimation process that still do not have solutions in the literature for encoding effort reduction.

On hardware design, one can find only a few works in the literature designing small solutions for VVC coding tools, mainly for transform coding. However, there is still space for design fast and low-power systems for both intra- and inter-frame prediction tools.

Additionally, since there is a lot of video content encoded with previous video coding standards such as H.264/AVC and HEVC the transcoding to VVC is necessary to enable gradual migration to the most recent video coding standard. Hence, there is a vast space to explore encoding effort reduction solutions for transcoding of H.264/AVC or HEVC to VVC format.

REFERENCES

AKIBA, T. et al. Optuna: A next-generation hyperparameter optimization framework. **Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining**, pp. 2623-2631, 2019.

ALSHIN, A., ALSHINA, E. Bi-directional optical flow for future video codec. **Data Compression Conference (DCC)**, pp. 83–90 2016.

BERGSTRA, J. et al. Algorithms for hyper-parameter optimization. **Neural Information Processing Systems Foundation (NIPS)**, v. 24, pp. 2546–2554, 2011.

BJONTEGAARD, G. **Calculation of average PSNR differences between RD Curves**, VCEG-M33, ITU-T SG16/Q6 VCEG, 13th VCEG Meeting: Austin, USA, Abril de 2001.

BOEHMKE, B., GREENWELL, B. **Hands on Machine Learning with R**. Chapman and Hall/CRC, 2019.

BOSSEN, F., et al. **VTM common test conditions and software reference configurations for SDR video**. JVET 20th Meeting, JVET-T2010, Oct. 2020.

BOSSEN, F. et al. **AHG report: Test Model Software Development (AHG3)**. JVET 23rd Meeting, JVET-W0003, Jul. 2021.

BROSS, B. et al. Overview of the Versatile Video Coding (VVC) Standard and its Applications. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, vol. 31, no. 10, pp. 3736-3764, Oct. 2021.

BUDAGAVI, M., FULDSETH, A., BJONTEGAARD, G. HEVC transform and quantization. **High Efficiency Video Coding (HEVC): Algorithms and Architectures**, Springer, pp. 141-169, 2014.

CHANG Y. et al. Multiple Reference Line Coding for Most Probable Modes in Intra Prediction. **Data Compression Conference (DCC)**, pp. 559-559, 2019.

CHEN, C. et al. Generalized bi-prediction method for future video coding. **Picture Coding Symposium (PCS)**, pp. 1–5, 2016.

CHEN, Y. et. al. An overview of core coding tools in the AV1 video codec. **Picture Coding Symposium (PCS)**, pp. 41–45, 2018.

CHEN, J., YE, Y., KIM, S. **Algorithm description for Versatile Video Coding and Test Model 10 (VTM 10)**. JVET 19th Meeting, JVET-S2002, Teleconference, Jul. 2020.

CHEN, F. et al. A fast CU size decision algorithm for VVC intra prediction based on support vector machine. **Multimedia Tools and Applications (MTAP),** 79, 27923–27939, Oct. 2020a.

CHEN, Y. et al. A novel fast intra mode decision for versatile video coding. **Journal of Visual Communication and Image Representation (JVCIR)**, v. 71, p. 102849, 2020b.

CHIANG, et al. **CE10.1: Combined and multi hypothesis prediction**. JVET, Ljubljana, SI, 11th meeting, document JVET-K0257, Jul. 2018.

CHIEN W. et al. Motion Vector Coding and Block Merging in the Versatile Video Coding Standard. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 31, n. 10, pp. 3848-3861, Oct. 2021.

CISCO. **Cisco Annual Internet Report (2018–2023) White Paper**. 2020. Available at: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html. Accessed on: Oct. 2021.

CORREA, G. et al. Encoding time control system for HEVC based on Rate-Distortion-Complexity analysis. **IEEE International Symposium on Circuits and Systems (ISCAS)**, pp. 1114-1117, 2015.

CORREA, G. et al. Online Machine Learning for Fast Coding Unit Decisions in HEVC. **Data Compression Conference (DCC)**, pp. 564-564, 2019.

COVER, T.; THOMAS, J. **Elements of Information Theory**. New York: John Wiley & Sons, 1991.

CUI, J. et al. Gradient-Based Early Termination of CU Partition in VVC Intra Coding. **Data Compression Conference (DCC)**, pp. 103-112, 2020.

DAEDE, T.; NORKIN, A.; BRAILOVSKIY, I. **Video Codec Testing and Quality Measurement**. draft-ietf-netvc-testing-08 (work in progress), p. 23, 2019.

DENG, X., XU, M. Complexity control of HEVC for video conferencing. **IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 1552-1556, 2017.

DE-LUXÁN-HERNÁNDEZ, S. et al. An Intra Subpartition Coding Mode for VVC. **IEEE International Conference on Image Processing (ICIP)**, pp. 1203-1207, 2019.

DE-LUXÁN-HERNÁNDEZ, S. et al. Design of the intra subpartition mode in VVC and its optimized encoder search in VTM. **Applications of Digital Image Processing XLIII**, v. 11510, p. 115100Y, 2020.

DHAPOLA, S. **COVID-19 impact: Streaming services to dial down quality as internet speeds fall**. Indian Express, 2020. Available at: https://indianexpress.com/article/technology/tech-news-technology/coronavirus-internet-speeds-slow-netflix-hotstar-amazon-prime-youtube-reduce-streaming-quality-6331237/. Accessed on: Oct 2021.

FAN, Y. et al. A Fast QTMT Partition Decision Strategy for VVC Intra Prediction. **IEEE Access**, v. 8, pp. 107900-107911, 2020.

FU, T. et al. Fast CU Partitioning Algorithm for H.266/VVC Intra-Frame Coding. **IEEE International Conference on Multimedia and Expo (ICME)**, pp. 55-60, 2019a.

FU, T., et al. Two-Stage Fast Multiple Transform Selection Algorithm for VVC Intra Coding. **IEEE International Conference on Multimedia and Expo (ICME)**, pp. 61-66, 2019b.

GAO, et al. Fast intra mode decision algorithm based on refinement in HEVC. **IEEE International Symposium on Circuits and Systems (ISCAS)**, pp. 517-520, 2015.

GAO, H. et al. Geometric Partitioning Mode in Versatile Video Coding: Algorithm Review and Analysis. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**. v. 31, n. 9, pp. 3603-3617, Sept. 2021a.

GAO, H., et al. Decoder-Side Motion Vector Refinement in VVC: Algorithm and Hardware Implementation Considerations. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 31, n. 8, pp. 3197-3211, Aug. 2021b.

GHANBARI, M. **Standard Codecs: Image Compression to Advanced Video Coding**. [S.l.]: Institution Electrical Engineers, 2003.

HALL, M. et al. The WEKA Data Mining Software: An Update. **ACM SIGKDD Explorations Newsletter**, v. 11, n. 1, pp. 10-18, Jan. 2009.

HE, Z. et al. Framework of AVS2-video coding. **IEEE International Conference on Image Processing (ICIP)**, pp. 1515–1519, 2013.

HM. **HEVC Test Model (HM)**. 2018. Available at: https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tags/HM-16.20. Accessed on: Oct. 2021.

HOSSIN, M., SULAIMAN, N. A Review on Evaluation Metrics for Data Classification Evaluations. **International Journal of Data Mining & Knowledge Management Process (IJDKP)**, v. 5, n. 2, p. 1, 2015.

HUANG, Y. et al. A VVC Proposal with Quaternary Tree Plus Binary-Ternary Tree Coding Block Structure and Advanced Coding Techniques. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 30, n. 5, pp. 1311-1325, May 2020.

HUANG, Y. et al. Block partitioning structure in the VVC standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 31, n. 10, pp. 3818-3833, Oct. 2021.

ITU-T. **HSTP-VID-WPOM – Working practices using objective metrics for evaluation of video coding efficiency experiments**. Technical Paper. Jul. 2020.

KARCZEWICZ M. et al. VVC In-Loop Filters. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 31, n. 10, pp. 3907-3925, Oct. 2021.

KE, G. et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. **Advances in Neural Information Processing Systems**, v. 30, pp. 3146-3154, 2017.

KIM, K. et al. MC Complexity Reduction for Generalized P and B Pictures in HEVC. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 24, n. 10, pp. 1723-1728, Oct. 2014.

KOEHRSEN, W. GitHub - WillKoehrsen/feature-selector. **Feature selector**. 2018. Available at: https://github.com/WillKoehrsen/feature-selector. Accessed on: Oct 2021.

KOO, M. et al. Low Frequency Non-Separable Transform (LFNST). **IEEE Picture Coding Symposium (PCS)**, pp. 1-5, 2019.

LAINEMA, J. et al. Intra Coding of the HEVC Standard. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 22, n. 12, pp. 1792-1801, Dec. 2012.

LEI, M. et al. Look-Ahead Prediction Based Coding Unit Size Pruning for VVC Intra Coding. **IEEE International Conference on Image Processing (ICIP)**, pp. 4120-4124, 2019.

LI, T. et al. DeepQTMT: A Deep Learning Approach for Fast QTMT-Based CU Partition of Intra-Mode VVC. **IEEE Transactions on Image Processing (TIP)**, v. 30, pp. 5377-5390, 2021a.

LI, Y. et al. Early Intra CU Size Decision for Versatile Video Coding Based on a Tunable Decision Model. **IEEE Transactions on Broadcasting (TBC)**, v. 67, n. 3, pp. 710-720, Sept. 2021b.

LIN, J., et al. Motion Vector Coding in the HEVC standard. **IEEE Journal of Selected Topics in Signal Processing (JSTSP)**, v. 7, n. 8, pp. 957-968, Dec. 2013.

LIU, X. et al. An Adaptive CU Size Decision Algorithm for HEVC Intra Prediction Based on Complexity Classification Using Machine Learning. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 29, n. 1, pp. 144-155, Jan. 2019.

LU, T. et al. Luma Mapping with Chroma Scaling in Versatile Video Coding. **Data Compression Conference (DCC)**, pp. 193-202, 2020.

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 (TCSVT)**, v. 13, n. 7, pp. 620-636, July 2003.

MARPE, D., WIEGAND, T. SULLIVAN, G. The H. 264/MPEG4 advanced video coding standard and its applications. **IEEE communications magazine**, v. 44, n. 8, p. 134-143, 2006.

MUKHERJEE, D. et al. The latest open-source video codec VP9-an overview and preliminary results. **Picture Coding Symposium (PCS)**, pp. 390–393, 2013.

NATEKIN, A.; KNOLL, A. Gradient Boosting Machines, a Tutorial. **Frontiers in Neurorobotics**, v. 7, pp. 21, 2013.

PFAFF, J. et al. Intra Prediction and Mode Coding in VVC. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 31, n. 10, pp. 3834-3847, Oct. 2021.

RICHARDSON, I. **The H.264 Advanced Video Compression Standard**. 2nd ed. Chichester: John Wiley and Sons, 2010.

ROSEWARNE, C. et al. **High Efficiency Video Coding (HEVC) Test Model 16 (HM 16)**. Document: JCTVC-V1002, Geneva, CH, Oct. 2015.

RUSSELL, S., NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Prentice Hall, 2002.

SALDANHA, M. et al. Complexity Analysis of VVC Intra Coding. **IEEE International Conference on Image Processing (ICIP)**, pp. 3119-3123, 2020a.

SALDANHA, M. et al. Fast Partitioning Decision Scheme for Versatile Video Coding Intra-Frame Prediction. **International Symposium on Circuits and Systems (ISCAS)**, pp. 1-5, 2020b.

SALDANHA, M. et al. Performance Analysis of VVC Intra Coding. **Journal of Visual Communication and Image Representation (JCVIR)**, v. 79, p. 103202, 2021a.

SALDANHA, M. et al. Configurable Fast Block Partitioning for VVC Intra Coding Using Light Gradient Boosting Machine. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, 2021b.

SALDANHA, M. et al. Learning-based Complexity Reduction Scheme for VVC Intra-Frame Prediction. **Visual Communications and Image Processing (VCIP)**, 2021c. (Accepted)

SALDANHA, M. et al. Fast Block Partitioning Scheme for Chrominance Intra Prediction of Versatile Video Coding Standard. **Journal of Electronic Imaging (JEI)**, v. 30, n. 5, p. 053009, 2021d.

SALDANHA, M. et al. Analysis of VVC Intra Prediction Block Partitioning Structure. **Visual Communications and Image Processing (VCIP)**, 2021e. (Accepted)

SALMON, R. et al. **Higher Frame Rates for more Immersive Video and Television**. 2011 British Broadcasting Corporation. Available at: http://www.bbc.co.uk/rd/publications/whitepaper209. Accessed on: Oct. 2021.

SAMUELSSON, J. Media Coding Industry Forum Progress Report. **SMPTE Motion Imaging Journal**, v. 129, n. 8, pp. 100-103, Sept. 2020.

SCHAFER, M. et al. An Affine-Linear Intra Prediction with Complexity Constraints. **IEEE International Conference on Image Processing (ICIP),** pp. 1089-1093, 2019.

SCHWARZ, H. et al. Improved Quantization and Transform Coefficient Coding for the Emerging Versatile Video Coding (VVC) Standard. **IEEE International Conference on Image Processing (ICIP)**, pp. 1183-1187, 2019.

SCHWARZ H. et al. Quantization and Entropy Coding in the Versatile Video Coding (VVC) Standard. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 31, n. 10, pp. 3891-3906, Oct. 2021.

SHARMAN, K.; SUEHRING, K. **Common Test Conditions**. JCT-VC 26th meeting, JCTVC-Z1100, 2017.

SHERMAN, W.; CRAIG, A. **Understanding Virtual Reality**: Interface, Application, and Design. Second Edition. Morgan Kaufmann Publishers, 2018.

SULLIVAN, G., WIEGAND, T. Rate-distortion optimization for video compression. **IEEE Signal Processing Magazine**, v. 15, n. 6, pp. 74 – 90, Nov. 1998.

SULLIVAN, G. et al. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 22, n. 12, pp. 1649-1668, Sep. 2012.

SULLIVAN, G. et al. Standardized Extensions of High Efficiency Video Coding (HEVC). **IEEE Journal of Selected Topics in Signal Processing (J-STSP)**, v. 7, n. 6, pp. 1001-1016, Dec. 2013.

TANG, N. et al. Fast CTU Partition Decision Algorithm for VVC Intra and Inter Coding. **IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)**, pp. 361-364, 2019.

TISSIER, A. et al. CNN Oriented Complexity Reduction of VVC Intra Encoder. **IEEE International Conference on Image Processing (ICIP)**, pp. 3139-3143, 2020.

VANNE, J. et al. Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 22, n. 12, p. 1885-1898, 2012.

VTM. **VVC Test Model (VTM)**. 2020. Available at: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/releases/VTM-10.0. Accessed on: Oct de 2021.

WIEN, M. **High efficiency video coding. Coding Tools and specification**. Signals and Communication Technology, Springer, v. 24, 2015.

XIPH. Xiph.org Video Test Media [derf's collection]. Xiph.org. Available at: https://media.xiph.org/video/derf/. Accessed on: Oct 2021.

XU, M. et al. Reducing Complexity of HEVC: A Deep Learning Approach. **IEEE Transactions on Image Processing (TIP)**, v. 27, n. 10, pp. 5044-5059, Oct. 2018.

YANG, H. et al. Low Complexity CTU Partition Structure Decision and Fast Intra Mode Decision for Versatile Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 30, n. 6, pp. 1668-1682, Jun. 2020.

ZHANG, K. et al. Enhanced cross-component linear model for chroma intra-prediction in video coding. **IEEE Transactions on Image Processing (TIP)**, v. 27, n. 8, p. 3983-3997, 2018.

ZHANG, K. et al. An Improved Framework of Affine Motion Compensation in Video Coding. **IEEE Transactions on Image Processing (TIP)**, v. 28, n. 3, Mar. 2019a.

ZHANG, L. et al. History-based Motion Vector Prediction in Versatile Video Coding. **IEEE Data Compression Conference (DCC)**, pp. 43-52, 2019b.

ZHANG, Z. et al. “Fast Adaptive Multiple Transform for Versatile Video Coding,” **IEEE Data Compression Conference (DCC)**, pp. 63-72, 2019c.

ZHANG, Q. et al. Fast CU Partition and Intra Mode Decision Method for H.266/VVC. **IEEE Access**, v. 8, pp. 117539-117550, 2020.

ZHAO, L. et al. Fast mode decision algorithm for intra prediction in HEVC. **IEEE Visual Communications and Image Processing (VCIP)**, pp. 1-4, 2011.

ZHAO, X. et al. Enhanced Multiple Transform for Video Coding. **Data Compression Conference (DCC)**, pp. 73-82, 2016.

ZHAO, X. et al. Transform Coding in the VVC Standard. **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 31, n. 10, pp. 3878-3890, Oct. 2021.

ZHAO, L. et al. Wide Angular Intra Prediction for Versatile Video Coding. **Data Compression Conference (DCC)**, pp. 53-62, 2019.

ZHAO, J., et al. Adaptive CU Split Decision Based on Deep Learning and Multifeature Fusion for H.266/VVC. **Scientific Programming Hindawi**, v. 2020, 1058-9244, Aug. 2020.

APPENDIX A – COMMON TEST CONDITIONS

The Common Test Conditions (CTC) for VVC were developed by the experts of JVET to conduct the experiments in a well-defined environment and to facilitate the comparison of results between the different techniques and tools developed. The video sequences specified by the CTC contain several distinct characteristics to provide a robust evaluation. Thus, the CTC is regularly updated with the goal of providing an evaluation that approximates the real environment of a video encoder.

The CTC defines four main test configurations to be used in the evaluations: (i) All-Intra (AI); (ii) Low-Delay (LD); (iii) Low-Delay P (LDP); and (iv) Random-Access (RA). The all-intra configuration defines that all frames of a video sequence are I-frames, where only the intra-frame prediction is available in the encoding process. In the low-delay, only the first frame of the encoded video sequence is I-frame and the remaining frames are B-frames, allowing inter-frame prediction with one or two references, whereas low-delay P allows only P-frames (one reference). The random-access configuration uses a hierarchical temporal structure of B-frames, where I- and B-frames are employed and the coding process is performed through GOPs.

For all configurations of this CTC, the encoder works with 10 bit-depth to represent each sample in the YUV format. This set of configuration allows the evaluation of encoding and decoding process for different scenarios. For instance, when a specific experiment is performed for a given intra coding tool, the all-intra configuration can be employed.

The video sequences defined for the experiments are divided into six classes, including video resolutions from 416×240 to 3840×2160 pixels, totalizing 22 video sequences. Classes A1 and A2 refer to six UHD 4K (3840×2160 resolution) video sequences. Class B has five video sequences with 1920×1080 resolutions. Class C and D represent videos with 832×480 and 416×240 resolutions, respectively, each one with four video sequences. Finally, Class E indicates three video sequences with 1280×720 resolution. Moreover, each video sequence should be encoded with 22, 27, 32, and 37 QP values.

The specifications of these video sequences are listed below:

Class	Video Sequence	Frames	Frame Rate	Bit Depth
A1	Tango2	294	60	10
	FoodMarket4	300	60	10
	Campfire	300	30	10
A2	CatRobot	300	60	10
	DaylightRoad2	300	60	10
	ParkRunning3	300	50	10
B	MarketPlace	600	60	10
	RitualDance	600	60	10
	Cactus	500	50	8
	BasketballDrive	500	50	8
	BQTerrace	600	60	8
C	BasketballDrill	500	50	8
	BQMall	600	60	8
	PartyScene	500	50	8
	RaceHorsesC	300	30	8
D	BasketballPass	500	50	8
	BQSquare	600	60	8
	BlowingBubbles	500	50	8
	RaceHorses	300	30	8
E	FourPeople	600	60	8
	Johnny	600	60	8
	KristenAndSara	600	60	8

APPENDIX B – BACKGROUND ON MACHINE LEARNING

Machine learning is a subfield of artificial intelligence concerned with designing algorithms capable of obtaining knowledge from data analysis. Algorithms based on machine learning are used to determine the value of dependent variables by looking at the value of some features in the dataset, identifying regularities, and building generalization rules that can be expressed as models (RUSSEL; NORVIG, 2002).

Supervised learning is an approach to building machine learning algorithms based on a training dataset containing the inputs and the desired outputs. This approach can be further categorized into regression and classification algorithms. The classification algorithms classify the input samples into discrete values (categories or classes). In machine learning, there are binary and multiclass classification algorithms; the first classifies the data into two classes, and the last classifies the data without restricting the number of classes. Binary classification algorithms are the most studied case since binary problems are easier to interpret and train. Besides, a multiclass problem can be reduced to an ensemble of several binary problems. This work uses supervised learning applying binary classification algorithms.

In most cases, the training process is performed offline using the entire training dataset. If the model provides a good generalization, a high performance is expected when tested with a dataset different from the training one. The performance of binary classification models can be evaluated using different metrics and most of them are based on known concepts in machine learning, such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FP). TP refers to the number of correctly classified positive values, whereas TN indicates the number of correctly classified negative values. In addition, FP and FN are interpreted as the additive inverse of TP and TN, respectively.

The metrics used in this work are accuracy and F1-score. Accuracy measures the ratio of correct predictions over the total number of instances evaluated (i.e., the sum of TP and TN divided by the sum of TP, TN, FP, and FN). F1-score is the harmonic mean between precision and recall values. Precision measures how many positive predictions are correct (i.e., TP divided by the sum of TP and FP), and recall measures how many positive cases the model correctly predicted over all the positive cases in the dataset (i.e., TP divided by the sum of TP and FN) (HOSSIN; SULAIMAN, 2015).

APPENDIX C – LIST OF PUBLICATIONS DURING THIS PHD

Papers Published in Journals Directly Related with this Thesis

1. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Configurable Fast Block Partitioning for VVC Intra Coding Using Light Gradient Boosting Machine. IEEE Transactions on Circuits and Systems for Video Technology, v. pp, p. 1-1, 2021.

2. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Performance Analysis of VVC Intra Coding. Journal of Visual Communication and Image Representation, v. 79, p. 103202, 2021.

3. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Fast Block Partitioning Scheme for Chrominance Intra Prediction of Versatile Video Coding Standard. Journal of Electronic Imaging, v. 30, p. 1, 2021.

4. Marcel Corrêa, Mário Saldanha, Alex Borges, Guilherme Corrêa, Daniel Palomino, Marcelo Porto, Bruno Zatt, Luciano Agostini. AV1 and VVC Video Codecs: Overview on Complexity Reduction and Hardware Design. IEEE Open Journal of Circuits and Systems, v. 2, p. 564-576, 2021.

Papers Published in Conferences Directly Related with this Thesis

1. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Fast Transform Decision Scheme for VVC Intra-Frame Prediction Using Decision Trees. IEEE International Symposium on Circuits and Systems, 2022. (Submitted)

2. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Analysis of VVC Intra Prediction Block Partitioning Structure. Visual Communications and Image Processing, 2021. (Accepted)

3. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Learning-based Complexity Reduction Scheme for VVC Intra-Frame Prediction. Visual Communications and Image Processing, 2021. (Accepted)

4. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Complexity Analysis of VVC Intra Coding. IEEE International Conference on Image Processing, 2020.

5. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Fast Partitioning Decision Scheme for Versatile Video Coding Intra-Frame Prediction. IEEE International Symposium on Circuits and Systems, 2020.

6. Mário Saldanha, Marcel Corrêa, Guilherme Corrêa, Daniel Palomino, Marcelo Porto, Bruno Zatt, Luciano Agostini. An Overview of Dedicated Hardware Designs for State-of-the-Art AV1 and H.266/VVC Video Codecs. IEEE International Conference on Electronics, Circuits and Systems, 2020.

Book Chapter

1. Vladimir Afonso, Mário Saldanha, Ruhan Conceição, Murilo Perleberg, Marcelo Porto, Bruno Zatt, Altamiro Susin, Luciano Agostini. Real-time architectures for 3D video coding. *VLSI Architectures for Future Video Coding*. 1ed.: Institution of Engineering and Technology, v. 1, p. 191-226, 2019.

Other Papers Published in Journals

1. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Fast 3D-HEVC Depth Map Encoding Using Machine Learning. IEEE Transactions on Circuits and Systems for Video Technology, v. 30, p. 850-861, 2020.

2. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Tile Adaptation for Workload Balancing of 3D-HEVC Encoder in Homogeneous Multicore Systems. IEEE Transactions on Circuits and Systems I-Regular Papers, v. 67, pp. 1704-1714, 2020.

3. Mário Saldanha, Ruhan Conceição, Vladimir Afonso, Giovani Ávila, Altamiro Susin, Marcelo Porto, Bruno Zatt, Guilherme Corrêa, Luciano Agostini. Complexity and compression efficiency assessment of 3D-HEVC encoder. Multimedia Tools and Applications, v. 79, pp. 25723-25746, 2020.

4. Gustavo Sanchez, Mário Saldanha, Ramon Fernandes, Rodrigo Cataldo, Luciano Agostini, César Marcon. 3D-HEVC Bipartition Modes Encoder and Decoder Design Targeting High-Resolution Videos. IEEE Transactions on Circuits and Systems I-Regular Papers, v. 67, p. 415-427, 2020.

5. Vladimir Afonso, Ruhan Conceição, Mário Saldanha, Luciano Braatz, Murilo Perleberg, Guilherme Correa, Marcelo Porto, Luciano Agostini, Bruno Zatt, Altamiro

Susin. Energy-Aware Motion and Disparity Estimation System for 3D-HEVC With Run-Time Adaptive Memory Hierarchy. IEEE Transactions on Circuits and Systems for Video Technology, v. 29, pp. 1878-1892, 2019.

6. Gustavo Sanchez, Mário Saldanha, Luciano Agostini, César Marcon. Analysis of parallel encoding using tiles in 3D High Efficiency Video Coding. Signal Image and Video Processing, v. 13, pp. 1079-1086, 2019.

7. Mariana Ucker, Vladimir Afonso, Mário Saldanha, Luan Audibert, Ruhan Conceição, Altamiro Susin, Marcelo Porto, Bruno Zatt, Luciano Agostini. High-Throughput Hardware for 3D-HEVC Depth-Map Intra Prediction. IEEE Design & Test, v. 37, pp. 7-14, 2020.

Other Papers Published in Conferences

1. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. TITAN: Tile Timing-Aware Balancing Algorithm for Speeding Up the 3D-HEVC Intra Coding. IEEE International Symposium on Circuits and Systems, 2019.

2. Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Time Reduction on 3D-HEVC Depth Maps Coding using Static Decision Trees Built Through Data Mining. XXV Simpósio Brasileiro de Sistemas Multimídia e Web, 2019.

3. Christopher Moura, Mário Saldanha, Gustavo Sanchez, César Marcon, Luciano Agostini. Fast Intra Mode Decision for 3D-HEVC Depth Map Coding using Decision Trees. IEEE International Conference on Electronics, Circuits and Systems, 2020.