J. Vis. Commun. Image R. 79 (2021) 103202

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

journal homepage: www.elsevier.com/locate/jvci

Performance analysis of VVC intra coding☆

Mário Saldanha [a,*], Gustavo Sanchez [b], César Marcon [c], Luciano Agostini [a]

[a] Federal University of Pelotas, Pelotas, Brazil
[b] IF Farroupilha, Alegrete, Brazil
[c] Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil

ARTICLE INFO

Keywords:
Versatile video coding
VVC
Intra-frame coding
Performance analysis

ABSTRACT

This article presents a performance analysis of Versatile Video Coding (VVC) intra-frame prediction. VVC is the next generation of video coding standards, which has been developed to supply the demand of upcoming video applications. VVC brings several innovations and enhancements for the intra-frame prediction to improve the encoding efficiency. These improvements comprise larger block sizes, more flexible block partitioning, more angular intra-frame prediction modes, multiple transform selection, non-separable secondary transform, among others. This article provides a detailed description of these tools, discussing how they work together in the intra-frame coding flow to raise the compression performance. Moreover, this article presents encoding complexity, encoding usage distribution, and rate-distortion-complexity analyses of the intra-frame prediction tools over different quantization scenarios. Based on these analyses, this article provides support for future works focusing on VVC intra-frame coding, including complexity reduction, complexity control, and real-time hardware design.

1. Introduction

The continuous growth of digital video consumption over the internet, including streaming services such as Amazon Prime Video, YouTube, and Netflix, has generated enormous pressure on the available bandwidth capabilities. This is especially true in the COVID-19 pandemic context when video streaming increased a lot, forcing YouTube and Netflix to reduce video quality to decrease internet traffic and to support the current demand [1]. Additionally, the interest of the users and industry for the development of immersive video content applications, such as Ultra-High Definition (UHD), High-Dynamic Range (HDR), and 360-degree videos, also keep growing. Previous video coding standards such as Advanced Video Coding (AVC) [2] and High-Efficiency Video Coding (HEVC) [3] do not have a satisfactory performance to meet the current market requirements associated with the next video applications. Hence, this fact is creating the demand for next-generation video coding technologies with capabilities beyond the current standards.

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 Versatile Video Coding (VVC) [4] 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 the emerging applications with various types of video content. 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) [5] was established to define a clear and reasonable licensing model for VVC.

VVC brings 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. These improvements include larger block sizes, flexible block partitioning using a Quadtree with nested Multi-type Tree (QTMT) structure [6], a higher number of angular intra-frame prediction modes [7], Multiple Reference Line (MRL) intra-frame prediction [8], Affine Motion Compensation (AMC) [9], Multiple Transform Selection (MTS) [10], Luma Mapping with Chroma Scaling (LMCS) [11], extended maximum Quantization Parameter (QP), improved implementations of quantization and entropy coding, among others [7].

These new tools improved the encoding efficiency but raised the

☆ This paper has been recommended for acceptance by Dr Zicheng Liu.

* Corresponding author.
E-mail addresses: mrdfsaldanha@inf.ufpel.edu.br (M. Saldanha), gustavo.sanchez@iffarroupilha.edu.br (G. Sanchez), cesar.marcon@pucrs.br (C. Marcon), agostini@inf.ufpel.edu.br (L. Agostini).

https://doi.org/10.1016/j.jvcir.2021.103202
Received 17 December 2020; Received in revised form 30 April 2021; Accepted 27 June 2021
Available online 16 July 2021

encoder complexity expressively, boosting the development of works focusing on minimizing this complexity. These works include prediction of unnecessary block partition evaluations with fast Coding Unit (CU) decisions [12–15], simplification of the intra-frame prediction reducing the number of modes evaluated [15,16], and speedup of the transform selection [17]. Moreover, some works focus on real-time coding by designing specific hardware architectures for these new coding tools [18–20]. However, since VVC is a recent standard, there is still a vast space for research on efficient complexity reduction algorithms and hardware design exploration.

In this scenario, identifying the required computational effort and the relevance of each coding module is mandatory to investigate further research challenges and develop solutions for reducing the encoder complexity with minor impacts on the coding efficiency. Some works presented evaluations of VVC considering approaches including performance comparisons with prior standards [21–24], memory assessment [25], and complexity evaluation of encoding and decoding [26,27].

Topiwala et al. [21] and García-Lucas et al. [22] compared HEVC, VVC, and AOMedia Video 1 (AV1) [28] considering the bitrate compression performance – in terms of Bjontegaard Delta bitrate (BDBR) [29] – and encoding time of the reference software. The experiments were conducted using video sequences recommended by JVET for the Random Access (RA) encoder configuration. Both works demonstrated that VVC outperforms HEVC and AV1 in terms of coding efficiency at the cost of a significant increase in the encoding time. Sidaty et al. [23] presented a subjective evaluation between VVC and HEVC, considering High Definition – HD (1920 × 1080) and UHD (3840 × 2160) video resolutions. The results showed that even subjectively, VVC encoder outperforms HEVC significantly, especially for low bitrates. Moreover, for some cases, VVC can halve the bitrate with the same HEVC visual quality. Panayides et al. [24] proposed a performance evaluation of VVC and AV1 in the healthcare domain, considering video sequences composed of ultrasound, emergency scenery, and general-purpose images. This evaluation considered the compression performance through BDBR and a subjective video quality assessment performed by an experienced medical. The results evidenced that VVC outperforms AV1 regarding compression performance and visual quality in all domains.

Cerveira et al. [25] presented a memory assessment of VVC, considering an overall memory profiling and an inter-frame prediction specific analysis. The memory profiling was performed using Intel® VTune™ Amplifier Profiling Tool [30]. The overall analysis showed that VVC increases memory access significantly compared to HEVC and the inter-frame prediction remains the critical coding bottleneck. The inter-frame prediction analysis revealed that larger CU size and QTMT partitioning contribute to this increased memory requirement.

Pakdaman et al. [26] analyzed the VVC encoding and decoding computational complexity and memory requirements, using Intel® VTune™ as the profiling tool. The results demonstrated that motion estimation is the most time-consuming tool for Low-Delay (LD) and RA configurations, whereas transform and quantization attain the highest complexity for All-Intra (AI). For decoding complexity distribution, in-loop filters occupy the highest complexity for LD and AI configurations, whereas motion compensation and in-loop filters are the most time-consuming tools for RA. The memory requirements analysis reported that the VVC encoding and decoding use 30x and 3x more memory bandwidth than HEVC, respectively, due to the larger CU size, QTMT partitioning, and more coding modes.

Tissier et al. [27] analyze the VVC intra-frame prediction for specifying the upper limits of encoding complexity reduction. They defined three levels of complexity reduction opportunities, including partitioning, intra-frame prediction, and transform. The upper limits are obtained when the encoder can predict the best decision of that level. The results showed that the block partitioning prediction allows reducing the complexity by up to 99%. The complexity reduction can reach up to 64% and 54% for intra-frame prediction and transform, respectively.

Although the works [21–27] show several details and profiling of VVC, none of them presented a detailed analysis of each intra-frame coding tool. This article fills this gap by presenting an in-depth analysis of VVC intra-frame coding, including a detailed description of intra-frame coding tools an encoding complexity distribution of these tools according to block sizes, the encoding mode usage distribution regarding the available block sizes, and a rate-distortion-complexity evaluation of each intra-frame coding tool.

Correlating the computational effort and the encoding modes usage distribution with the impact on compression efficiency of each coding tool allows evaluating the performance of the new tools introduced in VVC intra coding. Furthermore, this performance evaluation provides support for the development of future works focusing on VVC intra coding.

This article extends our previous work [31], which analyzed the complexity distribution of coding tools in the VVC intra-frame prediction. Our novel contributions in this work are: (i) a dense description with relevant details of block partitioning and coding tools of the VVC intra-frame prediction; (ii) a detailed comparison of complexity and coding efficiency between VVC and HEVC; (iii) an encoding complexity distribution analysis, including primary and secondary transforms; (iv) a usage distribution analysis, considering block sizes, prediction modes, and transforms; (v) a rate-distortion-complexity evaluation to verify the impact in complexity reduction and compression efficiency when disabling intra coding tools; and (vi) a general discussion about complexity reduction possibilities for VVC intra-frame coding.

2. VVC Block Partitioning Structure

VVC follows a block-based hybrid video coding approach, an underlying concept of all major video coding standards such as AVC and HEVC. In this concept, each frame of a video sequence is split into blocks, and all blocks are processed in sequence by intra- and inter-frame prediction, forward/inverse transform and quantization, and entropy coding.

Since block partitioning plays an essential role in compression efficiency, 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 ultra-high video resolutions. Each input frame is partitioned into Coding Tree Units (CTUs) with up to 128 × 128 samples, whereas HEVC allows CTUs with a maximum size of 64 × 64 samples. Additionally, each CTU can be recursively partitioned into smaller blocks referred to as Coding Unit (CU).

VVC adopts the same concept of Quadtree (QT) used in the HEVC and introduces the Multi-Type Tree (MTT) partitioning structure, enabling rectangular-shaped CU sizes through Binary Tree (BT) and Ternary Tree (TT) [7]. These combined structures (QT + MTT) are named QTMT and allow six types of partitions shown in Fig. 1. A CU can be defined as no split, and the coding process is carried out with the current CU size. Otherwise, this CU is split using a quad, binary, or ternary tree structure. The quadtree structure splits a CU into four equal-sized square sub-CUs. The binary tree splits a CU into two symmetric sub-CUs. The ternary tree splits a CU into three sub-CUs, composed of a central one and two sides of sub-CUs, having 50% and 25% of the original CU size, respectively. Finally, BT and TT can be performed in vertical and horizontal directions.

Fig. 2 presents the QTMT structure for a 128 × 128 CTU split into several CUs with different QT and MTT levels; black lines represent QT splitting, green lines denote binary splitting, and red lines indicate ternary splitting. 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 [3].

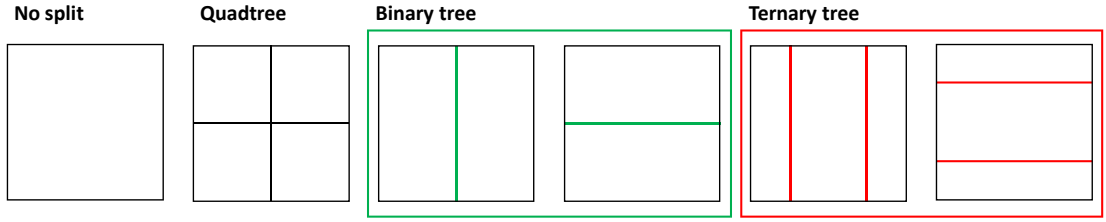


Fig. 1. Partition types of the QTMT partitioning structure.

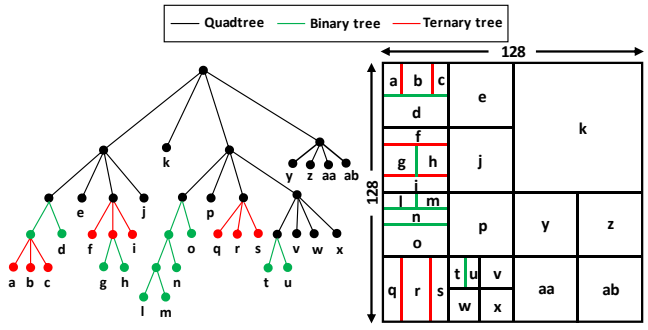


Fig. 2. Example of the QTMT partitioning structure with tree and block representations.

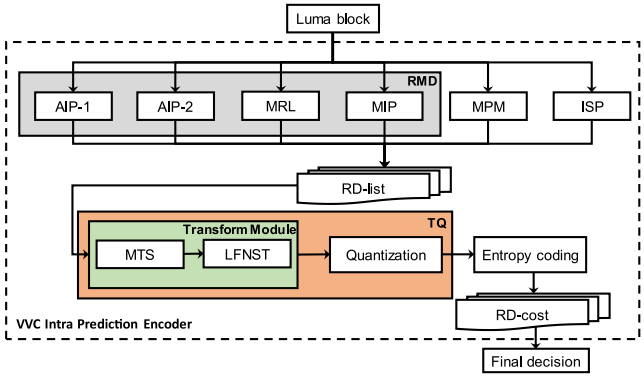


Fig. 3. Encoding flow of the VVC intra-frame prediction for luminance blocks.

Firstly, the QTMT structure split each CTU recursively in a QT structure; then, the QT leaf nodes can be further partitioned in an MTT structure. MTT can split a CU recursively through BT and TT partitions. However, it is important to highlight that once a CU is partitioned using the MTT structure, no further QT partitioning is allowed. The size of CUs may be as large as the CTU size (maximum size of 128 × 128 samples) or as small as 4 × 4 samples, encompassing square- and rectangular-shaped blocks. The maximum depth levels of QT and MTT structure are five and three, respectively, and the minimum luminance CU sizes obtained with QT and MTT are 8 × 8 and 4 × 4 samples, respectively. Regarding chrominance, the maximum and minimum CU sizes are 64 × 64 and 4 × 4 samples, respectively [7].

Only for I-slices, VVC allows partitioning chrominance blocks separately from the luminance blocks [7]. In other words, for I-slices, a chrominance block does not necessarily have the same coding tree structure obtained for the correlated luminance block, but the partition process is the same for luminance and chrominance samples regarding all other slice types (P and B) [7]. The intra-frame prediction is performed with square and rectangular CU sizes of 4 × 4 up to 64 × 64 (maximum transform size) for luminance blocks and 4 × 4 up to 32 × 32 samples for chrominance blocks. Besides, VVC allows chrominance blocks having 8 × 2, 16 × 2, and 32 × 2 samples.

QT with nested MTT partition structure enables high flexibility of representing block sizes and shapes. Hence, the VVC encoder allows more flexible block partition types that adapt to a wide range of video characteristics, resulting in better compression performance. However, the computational effort is expressively increased since this flexibility requires that the encoder considers all splitting possibilities.

3. VVC Intra-Frame Prediction

VVC intra-frame prediction introduces innovations to improve the encoding performance further. Fig. 3 presents the VVC intra-frame encoding flow for luminance samples; the encoder evaluates several encoding modes to minimize the Rate-Distortion (RD) cost. Thus, the encoding process selects the prediction mode that reaches the lowest RD-cost.

Similar to the HEVC Test Model (HM) [32], the intra-frame prediction of the VVC Test Model (VTM) [33] employs Rough Mode Decision (RMD) and Most Probable Modes (MPM) [34] 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 Rate-Distortion Optimization (RDO), which involves more complex operations, implying a prohibitive computational effort.

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. After, 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.

VVC brings novel intra-frame coding modes compared to HEVC, including Angular Intra Prediction-2 (AIP-2), MRL, Matrix-based Intra Prediction (MIP) [35], and Intra Subpartition (ISP) [36]. The Angular Intra Prediction-1 (AIP-1) in Fig. 3 represents the same HEVC intra-frame prediction modes. After processing these prediction tools, the modes inserted into the RD-list are processed by the residual coding, which includes Transform and Quantization (TQ) steps. Subsequently, the RD-costs are obtained applying Entropy Coding (EC) in the TQ flow results. The transform module encompasses MTS and Low-Frequency Non-Separable Transform (LFNST) [37].

VVC inherits the HEVC prediction modes and inserts Cross-Component Linear Model Prediction (CCLM) for chrominance blocks [38], where chrominance samples are predicted based on the reconstructed luminance samples by using a linear model. The encoding flow of chrominance blocks are not addressed in this work, but more details can be found in [7].

A. 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. The Planar and DC modes remain with the same approach used in HEVC. Fig. 4 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. Although Planar and DC are non-angular prediction modes, we call this tool of Angular Intra Prediction (AIP) for simplifying.

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 Fig. 4) and inserts a few modes into the RD-list. AIP-2 performs a refinement step to evaluate 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 [7].

VVC allows the intra-frame prediction of rectangular blocks due to the CU shapes obtained in the QTMT partitioning structure; it applies Wide-Angular Intra Prediction (WAIP) [39] since the angles of the 35-conventional angular modes were developed targeting square 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.

Additionally, for rectangular blocks, the DC prediction mode considers only the larger block side samples to provide a computationally efficient implementation.

B. Multiple Reference Line (MRL)

MRL [8] allows the use of more reference lines for the VVC intra-frame prediction than the ones used in HEVC. Fig. 5 shows a block size of 4 × 4 samples and the reference lines used in VVC intra-frame prediction when MRL is enabled. Reference 1 (index 0) refers to the nearest reference line, and it 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.

MRL 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, MRL evaluates only six MPMs for the two extra reference lines (reference lines 2 and 3) [8].

C. Matrix-based Intra Prediction (MIP)

MIP [35] is an alternative approach to the conventional angular intra-frame prediction modes. MIP performs the intra-frame prediction through matrix multiplication and samples interpolation. Fig. 6 demonstrates the MIP process for a block of 8 × 8 samples, where neighboring samples of the adjacent reference lines are also used as input for the prediction. These neighboring samples are subsampled to perform the matrix multiplication, followed by the addition of an offset (b_k) and a linear interpolation to obtain the predicted block [35]. A set of matrices were defined according to the block size by offline training through neural networks, and each matrix represents a prediction mode.

MIP allows 16 matrices for 4 × 4 blocks, eight matrices for blocks with width and height less than or equal to eight, and six matrices for the remaining block sizes (i.e., width and height larger than eight). Besides, the transposed matrices are also considered, doubling the number of prediction modes for each set. These prediction modes also are evaluated using RMD, and the RD-list is updated with the lowest SATD-based costs among AIP, MRL, and MIP prediction modes [35].

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

D. Intra Subpartition (ISP)

ISP [36] explores the correlations among block samples to improve the VVC intra-frame prediction. ISP divides the current block horizontally or vertically into subpartitions sequentially encoded using the same intra-frame prediction mode. The subpartitions are processed from top to bottom (horizontal split) or left to right (vertical split). The reconstructed samples of each encoded subpartition are used as reference samples for the next subpartition, increasing the reference sample

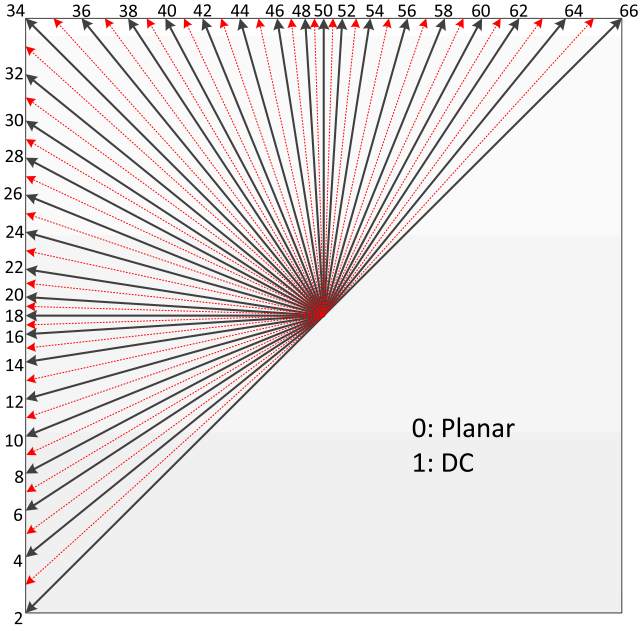


Fig. 4. VVC angular intra-frame prediction modes [7].

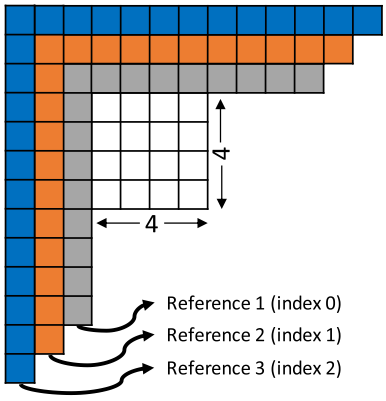


Fig. 5. Illustration of MRL intra prediction.

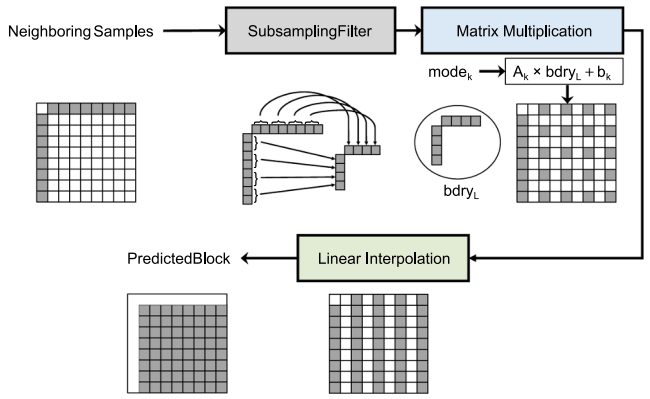


Fig. 6. Matrix-based intra prediction flow for a block size of 8 × 8 samples.

correlation compared to the conventional approach, which locates the reference samples at the left and above boundaries of the predicting block. Fig. 7 exemplifies ISP for a 16 × 16 block split into 16 × 4 horizontal and 4 × 16 vertical subpartitions.

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

ISP cannot use the RMD process since it requires the real reconstructed samples used as a reference to get the next subpartition 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 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.

E. Multiple Transform Selection (MTS)

MTS [10] enhances the VVC residual coding by including Discrete Cosine Transform VIII (DCT-VIII) and Discrete Sine Transform VII (DST-VII) beyond DCT-II used as the main transform in HEVC. Like HEVC, VVC applies DCT-II for both horizontal and vertical directions. DST and DCT families allow using separate transforms in horizontal and vertical directions when MTS is enabled [10]. For instance, applying DST-VII in the horizontal direction allows the use of DCT-VIII or DST-VII for the vertical direction, and vice versa. These transforms can be applied for square- and rectangular-shaped blocks. Combinations of DCT-II and DST-VII are performed only for the ISP tool based on the subpartition size [7].

The DCT-II transform may have block sizes ranging from 4 × 4 to 64 × 64 samples, whereas DCT-VIII and DST-VII may have blocks ranging from 4 × 4 to 32 × 32 samples. High-frequency coefficients are zeroed out for transforming blocks of sizes equal to 64 (width or height) and 32 for DCT-VIII and DST-VII to decrease the computational complexity. Thus, only low-frequency coefficients are retained.

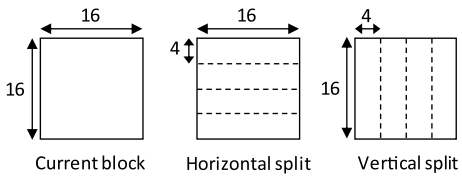


Fig. 7. Intra subpartition for a 16 × 16 block split into 16 × 4 horizontal and 4 × 16 vertical subpartitions.

Finally, using the same approach of HEVC, 32 × 32 residual or smaller blocks can be encoded with the Transform Skip Mode (TSM) that processes the residue signal in the quantization step. Then, VVC intra-frame prediction evaluates DCT-II (MTS index 0), TSM (MTS index 0), and DST-VII/DCT-VIII (MTS index 1) through RDO.

F. Low-Frequency Non-Separable Transform (LFNST)

LFNST [37] is a secondary transform of the VVC intra-frame prediction that further decorrelates the low-frequency primary transforms coefficients. LFNST is composed of two secondary transform sets with four non-separable transform matrices for each set[37]. The transform matrix evaluated for each set is defined based on the intra-frame prediction mode [7]. Thus, LFNST enables to evaluate (i) only primary transform (LFNST index 0), (ii) transform-set one (LFNST index 1), and (iii) transform-set two (LFNST index 2); however, LFNST can be applied only when DCT-II is the primary transform.

G. Quantization

The quantization module processes the residual information of the encoding block 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, whereas higher QP values provide a higher compression rate at the cost of image quality losses [40]. 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 [7,41].

H. Entropy Coding (EC)

As AVC and HEVC, the entropy coding of VVC is based on the Context Adaptive Binary Arithmetic Coding (CABAC) [42] 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 performed some improvements, such as the multi-hypothesis probability update model, separate residual coding for transformed blocks and blocks encoded with TSM, and context modeling for transform coefficients [7]. 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.

4. VVC Intra-Frame Prediction Analysis

This section presents the VVC intra-frame prediction analysis. Subsection 4.A introduces the methodology used in the experiments, Subsection 4.B presents a complexity and compression performance evaluation between VVC and HEVC, Subsection 4.C displays the encoding complexity distribution of luminance and chrominance, Subsection 4.D analyzes the block size complexity and usage distribution, Subsection 4.E shows encoding modes complexity and usage distribution, and Subsection 4.F presents a rate-distortion-complexity evaluation of VVC intra-frame coding tools. Finally, Subsection 4.G presents a general discussion, indicating complexity reduction possibilities for VVC intra-frame coding.

A. Methodology

All the analyses presented in this article followed the Common Test Conditions (CTC) [43] for Standard Dynamic Range (SDR) video

sequences 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 in this text. The experiments for HEVC were executed in HM software (version 16.20) with simulations regarding the All-Intra encoder configuration. For both encoder configurations of VTM and HM, the default temporal subsampling factor of 8 was considered (i.e., the encoding process is performed every 8 frames).

The CTC specification has been developed to be a benchmark to evaluate coding tools and 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. Thus, CTC is regularly enhanced with the insertion of new videos, coding settings, or both. The latest 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.

The compression performance and complexity were measured using BDBR and encoding time, respectively. For obtaining the encoding complexity distribution and the usage distribution of the encoding tools, modifications and additional functions were performed in the VTM reference software.

Table 1 defines the acronyms used in this work to avoid ambiguities among the analyzed coding tools.

B. VVC vs HEVC Compression Performance and Complexity

The first experiment compares the compression performance and the encoding complexity of VTM and HM. Fig. 8 presents the compression efficiency of VTM for luminance (Y) and chrominance components (Cb and Cr) compared to HM, considering AI encoder configuration.

One can notice from Fig. 8 that VTM performs better than HM for all cases regarding video resolution and encoding component. 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, VVC can provide a much higher compression rate than HEVC for intra-frame prediction, especially for high-resolution videos.

Table 1
Definition of the acronyms used for the encoding tools.

Acronym	Definition
BT	Binary Tree
TT	Ternary Tree
AIP	Angular Intra Prediction
MRL	Multiple Reference Line
MIP	Matrix-based Intra Prediction
ISP	Intra Subpartition
MTS	Multiple Transform Selection
LFNST	Low-Frequency Non-Separable Transform
TSM	Transform Skip Mode
TQ	Transform and Quantization
EC	Entropy Coding

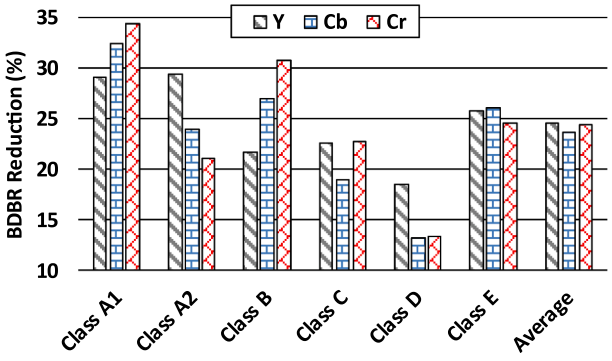


Fig. 8. Compression efficiency of VTM compared to HM for the AI configuration.

Fig. 9 displays the encoding complexity increase for each class of video sequences and QP values. One can notice that QP = 22 has the highest encoding complexity 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 coding complexity 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 complexity 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 complexity 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 bitrate. However, this efficiency requires a high computational effort, hampering the real-time video coding.

C. Complexity Distribution of Luminance and Chrominance

Fig. 10 displays the VVC intra-frame encoding complexity 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.

The luminance encoding has the highest complexity in VVC intra-frame coding, obtaining a maximum and minimum of 89% and 84%

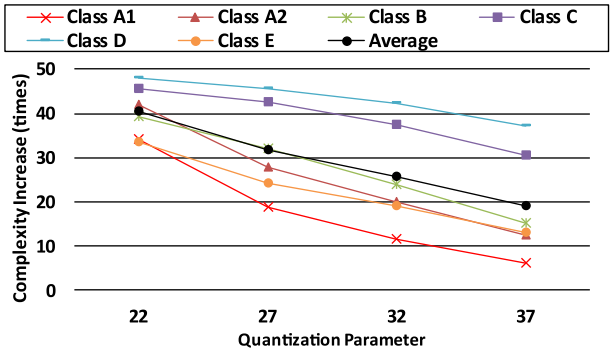


Fig. 9. Encoding complexity increase of VTM compared to HM for the AI configuration.

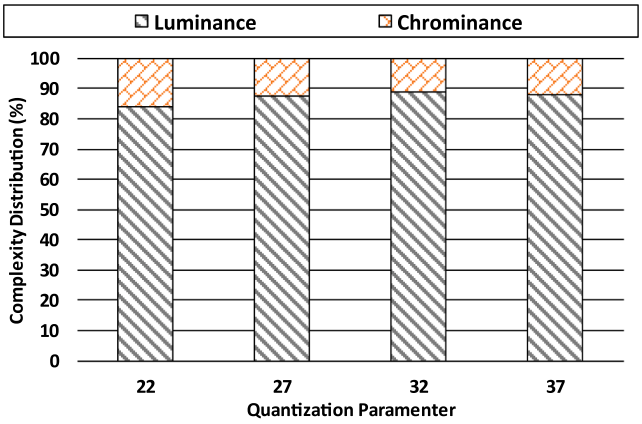


Fig. 10. Encoding complexity distribution for luminance and chrominance components considering the QP values defined in CTC.

of the total encoding complexity 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, whereas luminance still evaluates AIP, MRL, MIP, ISP and MTS coding tools. On average, luminance coding is 87% more complex than the chrominance coding in the AI scenario.

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

D. Block Size Analysis

Fig. 11 presents the average encoding time per frame and the usage distribution for each available luminance block size considering the QP corner cases; blue and orange bar denote the results for QP = 22 and QP = 37, respectively. The x-axis represents each block size, ordered from the largest block size 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 [7].

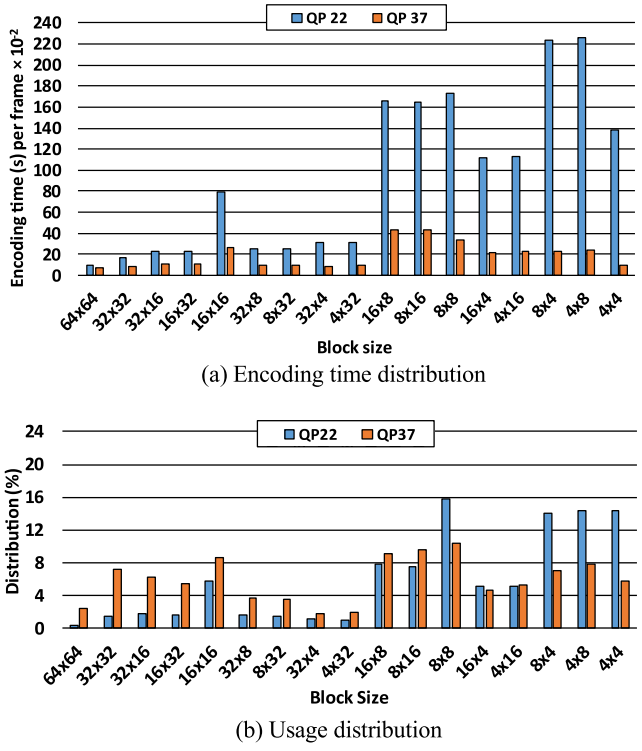


Fig. 11. (a) Encoding time distribution and (b) usage distribution for luminance block sizes considering QP = 22 and QP = 37.

From Fig. 11(a), one can notice that the most time-consuming blocks have sizes of 16 × 16, 16 × 8, or smaller for both QPs evaluated. However, the QP variation produces different encoding time distributions for each block size; lower QPs concentrate the encoding time distribution in the block sizes with smaller areas, whereas higher QPs have a more heterogeneous encoding time distribution.

Analyzing the average occurrence of each available block size in Fig. 11(b), one can notice that the block size selection also is highly dependent on the QP value; higher and lower QP values imply selecting larger and smaller block sizes, respectively. The block selection for QP = 22 concentrates in 16 × 8 samples or smaller sizes, with 84.0% of the occurrences. Blocks larger than 16 × 8 occur less than 2% of the time, except for the 16 × 16 block that occurs 5.7% of the time. 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 use 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.

E. Encoding Mode Analysis

Fig. 12 displays the complexity 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 complexity in all cases. Then, Fig. 12 omits part of the residual coding complexities to visualize better the other steps. Fig. 12(a) presents the encoding complexity with QP = 22 and Fig. 12(b) with QP = 37. Comparing both graphs, one can conclude that the residual coding complexity decreases for higher QPs; in this case, the prediction tools represent a higher percentage of the encoder complexity. 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 complexity (QP = 37). MRL and AIP-2 demand, together, less than 4.5% of the encoding complexity, on average, in both cases.

The high complexity of the residual coding is mainly noticed in the transform and quantization steps, demonstrating that MTS and LFNST evaluations increased the encoding complexity of transform process, whereas the quantizer indexes selection [41] raised the quantization step complexity. Additionally, 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, which also contributes to this high complexity.

Other conclusion when observing Fig. 12 is that as smaller is the block size, as higher tends to be the percentual effort spent in the prediction steps. This occurs, mainly, because 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 complexity for all cases assessed. It occurs because,

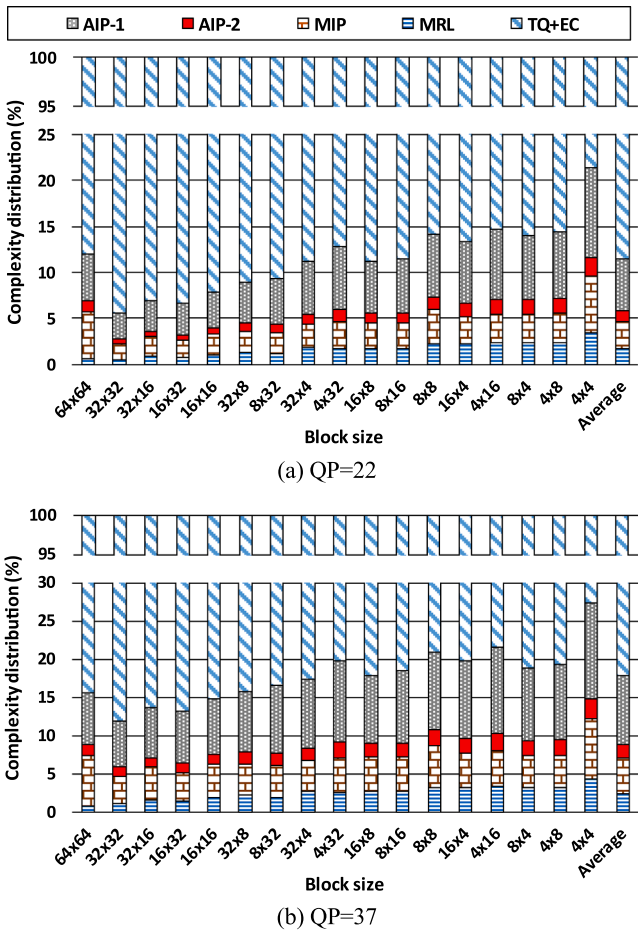


Fig. 12. Complexity distribution for the intra-frame prediction tools considering (a) QP = 22 and (b) QP = 37.

for each prediction mode in the RD-list, the residual coding flow is done several times, considering 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. Fig. 13 shows this analysis considering AIP-1, AIP-2, MRL, MIP, and ISP prediction modes. For both QP values in Fig. 13(a) and Fig. 13(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 block sizes 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 time, on average.

Since the transform step demand a high computational effort, the

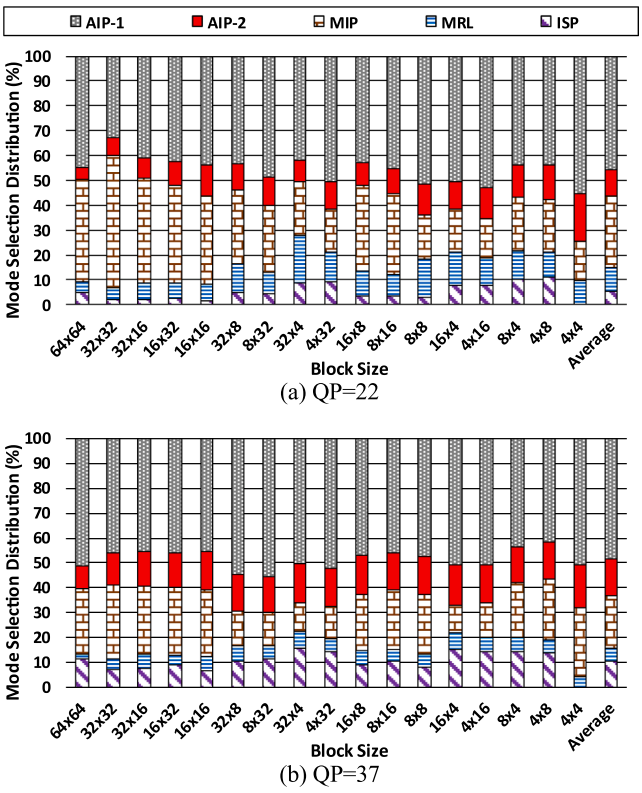


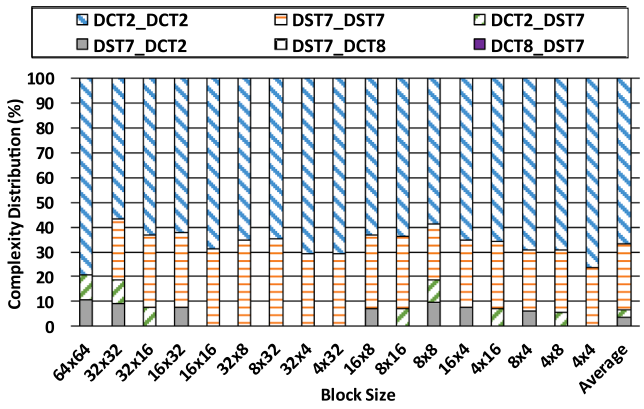
Fig. 13. Mode selection distribution for intra-frame prediction tools considering (a) QP = 22 and (b) QP = 37.

following analysis exhibits the encoding complexity distribution of primary transform combinations. Fig. 14 displays the average encoding complexity 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 complexity of DCT2_DCT2 also encompasses the encoding time of TSM.

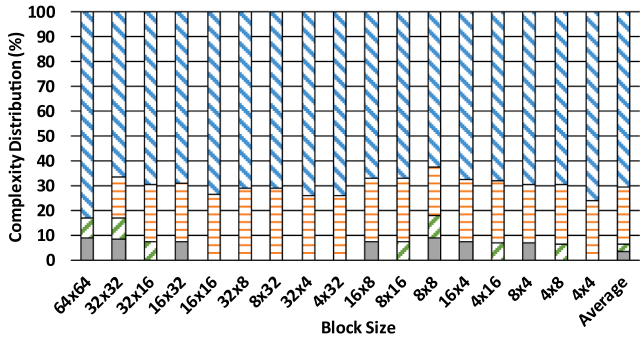
It is important to highlight that the DCT-II and DST-VII transforms can be combined only for the ISP predicted blocks and LFNST index 0 (i.e., without secondary transform). In this case, DST-VII is implicitly applied in the horizontal, vertical or both directions if the block width, height or both have 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 complexity, 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. DCT2_DCT2 is the most complex transform operation for both QPs, followed by DST7_DST7. Additionally, the remaining transform combinations represent less than 7% of complexity, on average.

Fig. 14(a) and Fig. 14(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 complexity. On average, for the corner QPs, DCT2_DCT2 and DST7_DST7 represent about 70% and 25% of the encoding complexity, respectively. The remaining

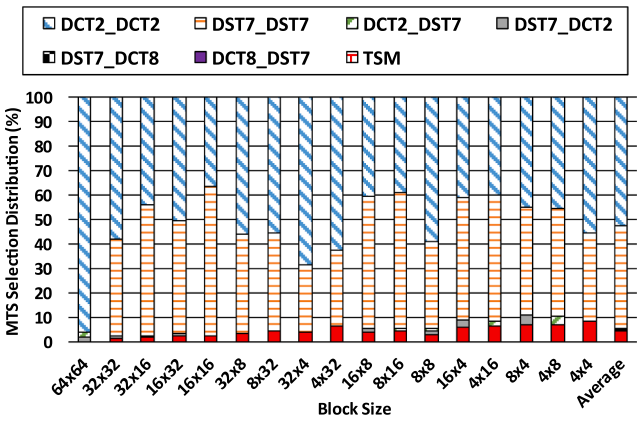


(a) QP=22

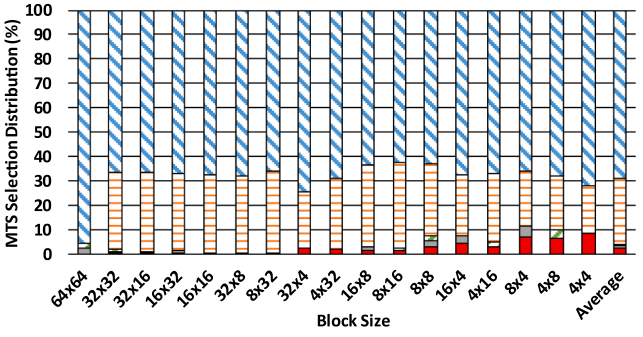


(b) QP=37

Fig. 14. Encoding complexity distribution for multiple primary transform selection considering (a) QP = 22 and (b) QP = 37.



(a) QP=22



(b) QP=37

Fig. 15. Selection distribution of multiple primary transforms considering (a) QP = 22 and (b) QP = 37.

transform combinations have less than 6.8% of the encoding complexity. For higher QPs, the DCT2_DCT2 combination tends to have a relative complexity 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 complexity 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. Fig. 15 presents this evaluation, considering the same transform combinations of the encoding complexity distribution shown in Fig. 14. 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, with a slight increase from 1.4% (QP = 22) to 1.6% (QP = 37). This occurs because these combinations are only allowed when ISP is used, and it 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.

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. LFNST allows the encoder to obtain satisfactory rate-distortion performance for most cases by evaluating only the DCT-II/TSM and DST-VII transforms. Thus, in the current implementation of VTM, the DCT-VIII transform combinations have a low potential to be chosen.

Fig. 16 illustrates the encoding complexity distribution among the LFNST encoding possibilities according to the block sizes and QP corner cases.

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 complexity, 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 complexity of both secondary transform evaluations (LFNST 1 and LFNST 2), these operations represent more than 55%

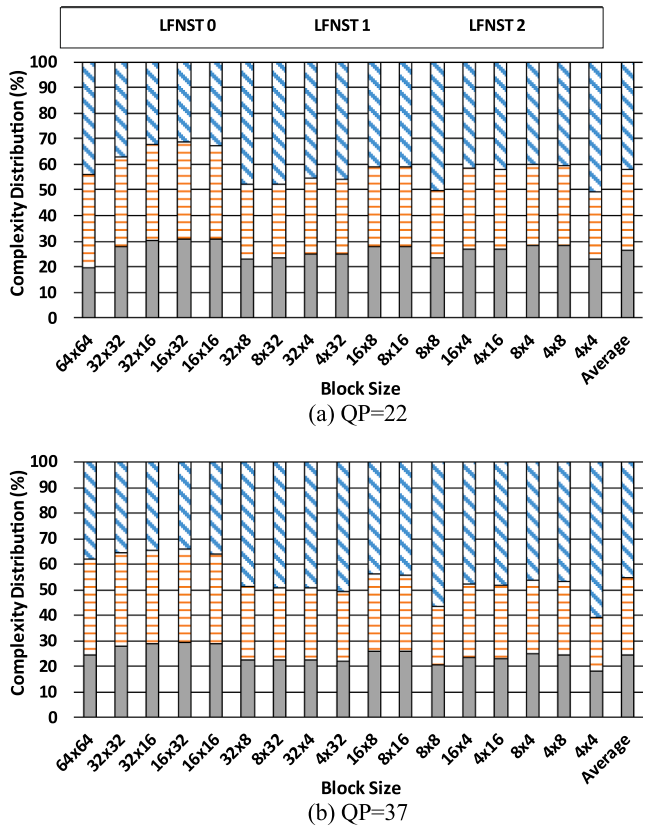


Fig. 16. Encoding complexity distribution of LFNST considering (a) QP = 22 and (b) QP = 37.

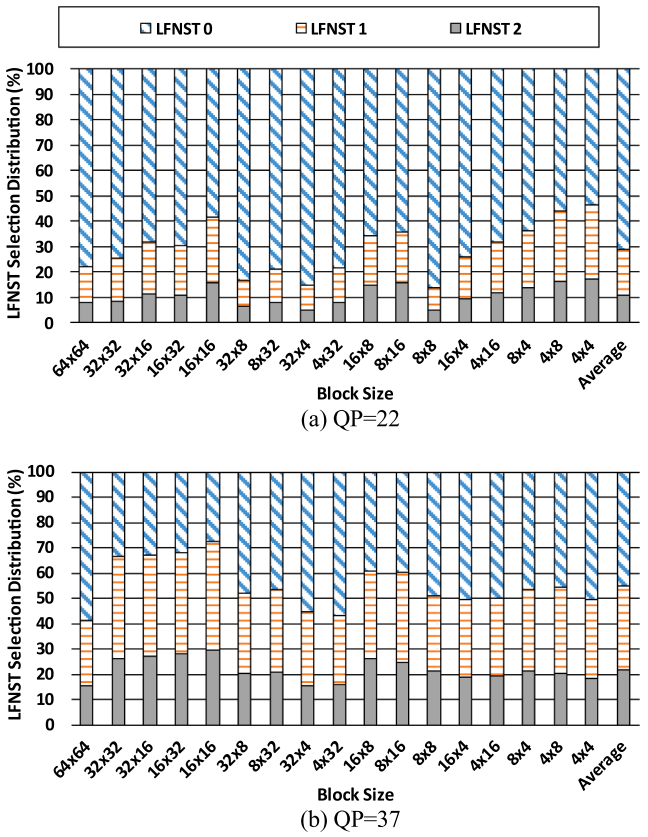


Fig. 17. LFNST selection distribution considering (a) QP = 22 and (b) QP = 37.

of the coding complexity, on average. Finally, observing Fig. 16, one can conclude that the complexity of LFNST does not have a direct correlation with the used QP value and block size.

Fig. 17 depicts the selection distribution of the secondary transform, also considering each available block size for the two corner QPs.

The secondary transform (LFNST 1 and 2) is less frequently used for lower QPs, being used 29.1% of the time with QP = 22 and 55.3% of the time 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. Fig. 17 displays that there is no clear correlation between the use of the secondary transform and the block size variation.

F. Rate-Distortion-Complexity of VVC Intra-Frame Coding Tools

This section presents a rate-distortion-complexity 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 AI 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 complexity reduction through BDBR and encoding time saving, respectively.

Table 2 presents the BDBR increase and complexity reduction 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. 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 complexity is reduced by 77.1% at

Table 2
Compression efficiency and complexity reduction when removing BT, TT, or both partitioning structures.

Class	BT less		TT less		BT + TT less	
	BDBR	CR	BDBR	CR	BDBR	CR
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%**

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 complexity 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 complexity reduction 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 class 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 have more detailed regions, which are better encoded with smaller block sizes, and BT and TT partition structures can enable block shapes adapted to these regions, providing a higher compression performance. This analysis shows that the QTMT partitioning structure provides significant compression performance gains while raising the VVC intra-frame coding complexity expressively.

Table 3 presents the average results of BDBR increase and complexity reduction when removing each intra-frame coding tool from the VTM intra-frame coding flow.

The highest complexity reduction results are obtained when

Table 3
Compression efficiency and complexity reduction when removing each VVC intra-frame coding tool.

Class	AIP-2 less		MRL less		MIP less	
	BDBR	CR	BDBR	CR	BDBR	CR
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	CR	BDBR	CR	BDBR	CR
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%**

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 in 1.2%.

Disable AIP-2 and MRL produce few gains in complexity reduction, attaining less than 1% of the encoding time reduction. 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 complexity is reduced by 11.3% at the cost of a BDBR increase of 0.6%.

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

G. General Discussion

All new intra-frame coding tools were added to the VVC specification because they improve the encoding efficiency expressively. Therefore, for scenarios requiring high coding efficiency these tools cannot be directly removed from the encoding flow to reduce the complexity; otherwise, the compression efficiency of VVC will be reduced drastically. However, based on the analyses presented in previous sections, several ideas and conclusions can be taken intending to elaborate efficient complexity reduction solutions for VVC intra-frame coding, beyond those solutions already inserted in the VTM.

Firstly, encoding small blocks such as 4×4, 4×8, 8×4, and 8×8 takes more time compared to 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 complexity presents slight variations among the evaluated QPs. Therefore, the QP can be considered when designing a complexity reduction 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 complexity 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 complex 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 complexity reduction results. A more limited complexity reduction can be obtained if considering the prediction tools. In this case, AIP-1 is the most time-consuming tool, and simplifying the RMD search step can also save the encoding time.

The residual coding is responsible for a considerable amount of encoding complexity to evaluate each intra-frame prediction mode through the TQ + EC flow for all available block sizes. This complexity 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 complexity for evaluated QPs. However, the analysis demonstrated that LFNST 1 and LFNST 2 are less frequently selected for a low QP, whereas the opposite happens for a high QP. 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 the predicting of the primary transform combination and when to use the secondary transform.

Besides, a VVC encoder may disable one or more tools depending on the requirements of complexity reduction and coding efficiency. Table 4 establishes some configurations indicating the tools to turn off, aiming the lowest impact on the coding efficiency when a determinate target of complexity reduction is required.

5. Conclusions

This article presented a detailed analysis of the intra-frame prediction in VVC - the next-generation of the video coding standard. To provide better compression efficiency, VVC brings several new tools and enhancements for the intra-frame coding. With these innovations, VVC enables a higher compression rate at the cost of a high computational effort compared to HEVC. This article presented the most relevant tools required by intra-frame prediction and showed their usage and encoding complexity distribution using the quantization scenarios defined by CTC. These analyses indicated that the QTMT block partitioning structure is responsible for an expressive increase in the encoding complexity and provides most of the compression rate gain. The residual coding is performed several times to define the best decision of primary and secondary transform, contributing to this increase in the encoding complexity.

This is the first article presenting an in-depth analysis of the VVC intra-frame prediction. The presented analysis is essential to understand the VVC intra-frame prediction behavior and support future solutions focusing on complexity reduction, complexity control, and real-time hardware designs.

Declaration of Competing Interest

The authors declare that they have no known competing financial

Table 4
Encoder configurations for six targets of complexity reduction and the estimated impact on the coding efficiency.

Complexity reduction	Tool off	BDBR
Target = 10%	ISP	0.5%
Target = 20%	MIP + ISP	~1.1%
Target = 30%	TT	1.2%
Target = 40%	TT	1.2%
Target = 50%	TT + ISP	~1.7%
Target = 70%	TT + MIP + ISP	~2.3%

interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors thank FAPERGS, CNPq and CAPES (Finance Code 001) Brazilian research support agencies that financed this investigation.

References

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

[2] D. Marpe, T. Wiegand, G.J. Sullivan, The H.264/MPEG4 advanced video coding standard and its applications, IEEE Commun. Mag. 44 (8) (Aug. 2006) 134–143, https://doi.org/10.1109/MCOM.2006.1678121.

[3] G.J. Sullivan, J. Ohm, W. Han, T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) Standard, IEEE Trans. Circuits Syst. Video Technol. (TCSVT) 22 (12) (Dec. 2012) 1649–1668, https://doi.org/10.1109/TCSVT.2012.2221191.

[4] ITU-T and ISO/IEC, Versatile Video Coding, ITU-T Rec. H.266 and ISO/IEC 23090-3, 2020.

[5] J. Samuelsson, Media Coding Industry Forum Progress Report, SMPTE Motion Imag. J 129 (8) (Sept. 2020) 100–103, https://doi.org/10.5594/JMI.2020.3002305.

[6] Y. Huang, et al., A VVC proposal with quaternary tree plus binary-ternary tree coding block structure and advanced coding techniques, IEEE Trans. Circuits Syst. Video Technol. (TCSVT) 30 (5) (May 2020) 1311–1325, https://doi.org/10.1109/TCSVT.2019.2945048.

[7] J. Chen, Y. Ye, S. Kim, Algorithm description for Versatile Video Coding and Test Model 10 (VTM 10), JVET 19th Meeting, JVET-S2002, Teleconference, Jul. 2020.

[8] B. Bross et al., CE3: Multiple reference line intra prediction (Test 1.1.1, 1.1.2, 1.1.3 and 1.1.4),“ JVET 12th Meeting, JVET-L0283, Macao, Oct, 2018.

[9] K. Zhang, Y. Chen, L. Zhang, W. Chien, M. Karczewicz, An Improved Framework of Affine Motion Compensation in Video Coding, IEEE Trans. Image Process. (TIP) 28 (3) (Mar. 2019) 1456–1469, https://doi.org/10.1109/TIP.2018.2877355.

[10] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li, W. Chien, Enhanced multiple transform for video coding, in: Data Compression Conf. (DCC), Snowbird, UT, 2016, pp. 73–82.

[11] T. Lu, et al., Luma mapping with chroma scaling in versatile video coding, in: Data Compression Conf. (DCC), Snowbird, UT, USA, 2020, pp. 193–202.

[12] A. Wieckowski, J. Ma, H. Schwarz, D. Marpe, T. Wiegand, Fast Partitioning Decision Strategies for The Upcoming Versatile Video Coding (VVC) Standard, in: IEEE Int. Conf. Image Process. (ICIP), Taipei, Taiwan, 2019, pp. 4130-4134, doi: http://dx.doi.10.1109/ICIP.2019.8803533.

[13] H. Yang, L. Shen, X. Dong, Q. Ding, P. An, G. Jiang, Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding, IEEE Trans. Circuits Syst. Video Technol. (TCSVT) 30 (6) (June 2020) 1668–1682, https://doi.org/10.1109/TCSVT.2019.2904198.

[14] T. Fu, H. Zhang, F. Mu and H. Chen, “Fast CU Partitioning Algorithm for H.266/VVC Intra-Frame Coding, in: IEEE Int. Conf. Multimedia Expo (ICME), Shanghai, China, 2019, pp. 55-60, doi: http://dx.doi.10.1109/ICME.2019.00018.

[15] Q. Zhang, Y. Wang, L. Huang, B. Jiang, Fast CU Partition and Intra Mode Decision Method for H.266/VVC, IEEE Access 8 (2020) 117539–117550, https://doi.org/10.1109/ACCESS.2020.3004580.

[16] Y. Chen, L. Yu, H. Wang, T. Li, S. Wang, A novel fast intra mode decision for versatile video coding, J. Vis. Commun. Image Represent. (2020), 102849.

[17] T. Fu, H. Zhang, F. Mu, H. Chen, Two-stage fast multiple transform selection algorithm for VVC intra coding, in: IEEE Int. Conf. Multimedia Expo (ICME), Shanghai, China, 2019, pp. 61-66, doi: http://dx.doi.10.1109/ICME.2019.00019.

[18] H. Azgin, E. Kalali, I. Hamzaoglu, An Approximate Versatile Video Coding Fractional Interpolation Hardware, in: IEEE Int. Conf. Consum. Electron. (ICCE), Las Vegas, NV, USA, 2020, pp. 1-4, doi: http://dx.doi.10.1109/ICCE46568.2020.9042986.

[19] Y. Fan, Y. Zeng, H. Sun, J. Katto, X. Zeng, A Pipelined 2D Transform Architecture Supporting Mixed Block Sizes for the VVC Standard, IEEE Trans. Circuits Syst. Video Technol. (TCSVT) 30 (9) (Sept. 2020) 3289–3295, https://doi.org/10.1109/TCSVT.2019.2934752.

[20] A. Kammoun, et al., Forward-Inverse 2D Hardware Implementation of Approximate Transform Core for the VVC Standard, IEEE Trans. Circuits Syst. Video Technol. (TCSVT) 30 (11) (Nov. 2020) 4340–4354, https://doi.org/10.1109/TCSVT.2019.2954749.

[21] P. Topiwala, M. Krishnan, W. Dai, Performance comparison of VVC, AV1, and HEVC on 8-bit and 10-bit content, SPIE Appl. Digit. Image Process. XLI 10752 (2018) 305–314, https://doi.org/10.1117/12.2322024.

[22] D. García-Lucas, G. Cebrián-Márquez, P. Cuenca, Rate-distortion/complexity analysis of HEVC, VVC and AV1 video codecs, Multimedia Tools Appl. (2020), https://doi.org/10.1007/s11042-020-09453-w.

[23] N. Sidaty, W. Hamidouche, O. Déforges, P. Philippe, J. Fournier, Compression Performance of the Versatile Video Coding: HD and UHD Visual Quality Monitoring, in: IEEE Picture Coding Symp. (PCS), Ningbo, China, 2019, pp. 1-5, doi: http://dx.doi.10.1109/PCS48520.2019.8954562.

[24] A.S. Panayides, M.S. Pattichis, M. Pantziaris, A.G. Constantinides, C.S. Pattichis, The Battle of the Video Codecs in the Healthcare Domain - A Comparative Performance Evaluation Study Leveraging VVC and AV1, IEEE Access 8 (2020) 11469–11481, https://doi.org/10.1109/ACCESS.2020.2965325.

[25] A. Cerveira, L. Agostini, B. Zatt, F. Sampaio, Memory assessment of versatile video coding, in: IEEE Int. Conf. Image Process. (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 1186–1190.

[26] F. Pakdaman, M.A. Adelimanesh, M. Gabbouj, M.R. Hashemi, Complexity analysis of next-generation VVC Encoding and decoding, in: IEEE Int. Conf. Image Process. (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 3134–3138.

[27] A. Tissier, A. Mercat, T. Amestoy, W. Hamidouche, J. Vanne, D. Menard, Complexity Reduction Opportunities in the Future VVC Intra Encoder, in: IEEE Int. Workshop Multimedia Signal Process. (MMSP), Kuala Lumpur, Malaysia, 2019, pp. 1–6.

[28] P. Rivaz, J. Haughton, AV1 Bitstream & Decoding Process Specification, Alliance for Open Media, 2018. Available: https://aomedia.org/av1-bitstream-and-decoding-process-specification/.

[29] G. Bjontegaard, Calculation of Average PSNR Differences between RDCurves, VCEG Meeting (2001) 1–5.

[30] Intel® VTune TM Profiler (Online). Available at: https://software.intel.com/en-us/vtune, access in Aug. 2020.

[31] M. Saldanha, G. Sanchez, C. Marcon, L. Agostini, Complexity Analysis of VVC Intra Coding, in: IEEE Int. Conf. Image Process. (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 3119–3123.

[32] HEVC Test Model (HM). Available at: https://vcgit.hhi.fraunhofer.de/jct-vc/HM, access in Aug. 2020.

[33] VVC Test Model (VTM). Available at: https://vcgit.hhi.fraunhofer.de/ jvet/VVCSoftware_VTM, access in Aug. 2020.

[34] L. Zhao, L. Zhang, S. Ma, D. Zhao, Fast mode decision algorithm for intra prediction in HEVC, in: IEEE Vis. Commun. Image Process. (VCIP), Tainan, Taiwan, 2011, pp. 1–4, doi: http://dx.doi.10.1109/VCIP.2011.6115979.

[35] M. Schäfer et al., An Affine-Linear Intra Prediction with Complexity Constraints, in: IEEE Int. Conf. Image Process. (ICIP), Taipei, Taiwan, 2019, pp. 1089-1093, doi: http://dx.doi.10.1109/ICIP.2019.8803724.

[36] S. De-Luxán-Hernández et al., An Intra Subpartition Coding Mode for VVC, in: IEEE Int. Conf. Image Process. (ICIP), Taipei, Taiwan, 2019, pp. 1203-1207, doi: http://dx.doi.10.1109/ICIP.2019.8803777.

[37] M. Koo, M. Salehifar, J. Lim, S. Kim, Low Frequency Non-Separable Transform (LFNST), in: IEEE Picture Coding Symp. (PCS), Ningbo, China, 2019, pp. 1–5, doi: http://dx.doi.10.1109/PCS48520.2019.8954507.

[38] K. Zhang, J. Chen, L. Zhang, X. Li, M. Karczewicz, Enhanced Cross-Component Linear Model for Chroma Intra-Prediction in Video Coding, IEEE Trans. Image Process. (TIP) 27 (8) (Aug. 2018) 3983–3997, https://doi.org/10.1109/TIP.2018.2830640.

[39] L. Zhao et al., Wide Angular Intra Prediction for Versatile Video Coding, 2019 Data Compression Conf. (DCC), Snowbird, UT, USA, 2019, pp. 53-62, doi: http://dx.doi.10.1109/DCC.2019.00013.

[40] M. Budagavi, A. Fuldseth, G. Bjontegaard, HEVC transform and quantization, High Efficiency Video Coding (HEVC): Algorithms and Architectures, Springer, pp. 141–169, 2014.

[41] H. Schwarz et al., Improved Quantization and Transform Coefficient Coding for the Emerging Versatile Video Coding (VVC) Standard, in: IEEE Int. Conf. Image Process. (ICIP), Taipei, Taiwan, 2019, pp. 1183–1187, doi: http://dx.doi.10.1109/ICIP.2019.8803768.

[42] D. Marpe, H. Schwarz, T. Wiegand, Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard, IEEE Trans. Circuits Syst. Video Technol. (TCSVT) 13 (7) (July 2003) 620–636, https://doi.org/10.1109/TCSVT.2003.815173.

[43] F. Bossen, J. Boyce, X. Suehring, X. Li, V. Seregin, JVET common test conditions and software reference configurations for SDR video. JVET 14th Meeting, JVET-N1010, 2019.

Mário Saldanha received the B.S. degree in Computer Science from the Federal University of Pelotas (UFPel), Pelotas, RS, Brazil, in 2016. In 2018, he received his M.Sc. degree in Computer Science from the UFPel. He is currently pursuing the Ph.D. degree in Computer Science at the UFPel. He is a member of the Video Technology Research Group (ViTech) at the UFPel. His research interests include complexity reduction and hardware-friendly algorithms for 2D/3D video coding.

Gustavo Sanchez is a Professor at the IFFarroupilha, Brazil, since 2014. Sanchez received the Electrical Engineer degree from the Sul-Rio-Grandense Federal Institute of Education, Science and Technology (2013) and B.S. degree in Computer Science from the Federal University of Pelotas (UFPel) (2012). In 2014, he received his M.Sc. degree in Computer Science from the UFPel. Sanchez obtained his Ph.D. degree in Computer Science at the Pontifical Catholic University of Rio Grande do Sul in 2019. He has more than 10 years of research experience on algorithms and hardware architectures for video coding. His research interests include complexity reduction algorithms, hardware-friendly algorithms and dedicated hardware design for 2D/3D video coding.

César Marcon is a Professor at the School of Computer Science of Pontifical Catholic University of Rio Grande do Sul (PUCRS), Brazil, since 1995. He received his Ph.D. in Computer Science from Federal University of Rio Grande do Sul, Brazil, in 2005. Professor Marcon is member of the Institute of Electrical and Electronics Engineers (IEEE) and of the Brazilian Computer Society (SBC). He is advisor of M.Sc. and Ph.D. graduate students at Graduate Program in Computer Science of PUCRS. Since 2005, prof. Marcon coordinated

nine research projects in areas of telecom, healthcare, and telemedicine. His research interests are in the areas of embedded systems in the telecom domain, MPSoC architectures, partitioning and mapping application tasks, fault-tolerance and real-time operating systems.

Luciano Agostini is a Brazilian Distinguished Researcher through a CNPq PQ-1D grant. He received the M.S. and Ph.D. degrees from Federal University of Rio Grande do Sul, Porto Alegre, Brazil, in 2002 and 2007 respectively. He is a Professor since 2002 at Federal University of Pelotas (UFPel), Brazil, where he leads the Video Technology Research Group (ViTech). He is advisor at the UFPel Master and Doctorate in Computer Science courses. He was the Executive Vice President for Research and Graduate Studies of UFPel from 2013 to 2017. He has more than 300 published papers in journals and conference proceedings. His research interests include 2D and 3D video coding, algorithmic optimization, arithmetic circuits and digital systems. Currently he is an Associated Editor of two IEEE CAS journals: TCSVT and OJCAS. Dr. Agostini is a Senior Member of IEEE and ACM. He is a member of the IEEE CAS, CS, and SPS societies. At the IEEE CAS, he is a member of the MSATC. He is also member of SBC and SBMicro Brazilian societies.