Performance Analysis of Depth Intra-Coding in 3D-HEVC

Gustavo Sanchez, Jarbas Silveira, *Member, IEEE*, Luciano V. Agostini, *Senior Member, IEEE*, and César Marcon, *Member, IEEE*

***Abstract*— The depth maps intra-frame prediction of 3D High Efficiency Video Coding (3D-HEVC) inherits all texture encoding techniques provided by HEVC and provides new coding tools for depth map predictions. These tools comprise algorithms, such as bipartition modes, intra-picture skip, and DC-only. This paper details these tools and shows how they work together with the original HEVC algorithms in the depth map intra-frame prediction for allowing high-efficiency encoding. Besides, this paper analyzes the encoding time and the encoding mode distribution of the intra-frame prediction tools over different quantization scenarios. We aim to provide support for upcoming works on depth map encoding, including complexity reduction and control, real-time embedded systems implementations, and even the development of improved tools to encode depth maps.**

***Index Terms*— 3D-HEVC, intra-frame prediction, depth maps, 3D video coding.**

I. Introduction

THREE-DIMENSIONAL (3D) videos allow the viewers to enjoy an experience beyond (two-dimensional) 2D videos through a depth perception of the scene. With the rising of the 3D device market by having applications such as 3D television [1] and free-viewpoint television [2] and with the increase of video resolution, there is an even higher demand for algorithms and standards, which provides quality and very high compression efficiency. In a high-resolution 2D video scenario, it is already a challenging process to deliver a video within a required bandwidth without a significant increase in the encoding time and energy consumption [3]. This problem is even worse for 3D videos because their production requires more than one camera [4]. The Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) [5] developed the 3D High Efficiency Video Coding (3D-HEVC) standard [6] to solve the compression rate problem.

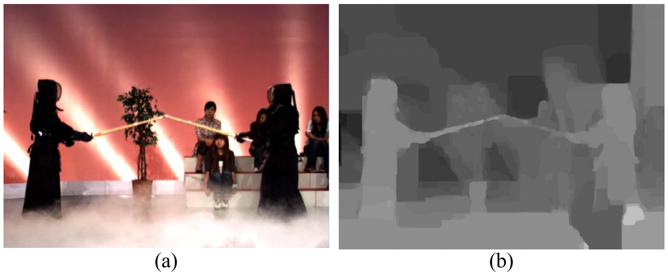


Fig. 1. Kendo video sequence frame with (a) texture view and (b) its associated depth map [11].

3D-HEVC adopts the Multiview Video plus Depth (MVD) representation [7] that associates a depth map to each texture frame. The depth maps are collocated with the texture image and represent the geometric distance among the objects and camera from the same viewpoint of the texture [8]. The MVD format allows the decoder to generate a dense set of virtual views located among transmitted views instead of requiring the transmission of this dense set of texture views [9].

View synthesis techniques such as Depth Image Based Rendering (DIBR) [10] interpolates texture frames based on depth maps. In general, view synthesis techniques are highly complex and time-consuming, which can be accelerated with the video cameras alignment. Therefore, 3D-HEVC has been designed specially to encode this set of aligned views.

Fig. 1 exemplifies a texture frame, and its associated depth map, extracted from Kendo video sequence [11]. One can notice the depth maps are characterized by containing large regions with similar values (presented in the background or object bodies) and sharp edges (presented in the object borders) [12], [13]. The depth map coding tools consider these characteristics to reach a high compression rate with a minimum distortion on the depth maps. The quality of the encoded depth maps is crucial to allow generating high quality synthesized views [14], which is a big challenge in a scenario, where higher resolution videos are required, and higher compression rates are desired.

Although depth maps are not displayed for the users, the quality of the encoded depth maps influences the video quality indirectly since most of the views displayed for the viewers are synthesized [15], [16]. The use of low-quality depth maps to generate virtual views produces distortions mainly around the places where there is a high depth variation

Manuscript received December 1, 2017; revised March 1, 2018; accepted July 31, 2018. Date of publication August 16, 2018; date of current version August 2, 2019. This work was supported in part by CNPq, in part by FAPERGS, and in part by CAPES Brazilian. This paper was recommended by Associate Editor J. Ostermann. *(Corresponding author: Gustavo Sanchez.)*

G. Sanchez is with the Department of Computer Science, IF Farroupilha, Alegrete 97555-000, Brazil, and also with Pontifical Catholic University of Rio Grande do Sul, Porto Alegre 90619-900, Brazil (e-mail: gustavo.sanchez@acad.pucrs.br).

J. Silveira is with the Engineering Laboratory in Computer Systems, Federal University of Ceará, Fortaleza 60020-181, Brazil (e-mail: jarbas@lesc.ufc.br).

L. V. Agostini is with the Group of Architectures and Integrated Circuits, Federal University of Pelotas, Pelotas 96010-610, Brazil (e-mail: agostini@inf.ufpel.edu.br).

C. Marcon is with the Department of Computer Science, Pontifical Catholic University of Rio Grande do Sul, Porto Alegre 90619-900, Brazil (e-mail: cesar.marcon@pucrs.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2018.2865645

on the views. Therefore, it is vital to encode the depth maps without smoothing the edges to avoid errors in the video synthesis [17]. Focusing on solving this problem, the JCT-3V experts included new coding tools for intra-frame prediction to preserve depth maps edges as efficiently as possible in the 3D-HEVC encoder, without affecting virtual views quality and obtaining a high compression rate. These tools include bipartition modes [18], intra-picture skip mode (also known as Depth Intra Skip - DIS) [19] and DC-only coding (also known as Segment-wise DC Coding - SDC) [20]. The bipartition modes are composed of Intra_Wedge and Intra_Contour, which are also known as Depth Modeling Mode 1 (DMM-1) and Depth Modeling Mode 4 (DMM-4), respectively.

The insertion of these tools raised the encoder complexity, boosting several works that propose to minimize this complexity. These works include simplifying HEVC intra prediction [21], speeding up the TQ and/or DC-only flows [22], [23], filtering the Intra_Wedge wedgelet list evaluation [21], [25], skipping the entire bipartition modes evaluation for homogeneous blocks [25]–[27], and avoiding the quadtree expansion [23], [28]. Besides, some works focus on real-time coding by designing specific hardware for these new encoding tools [29]–[31]. However, there is still a room for researches on efficient timesaving algorithms and hardware design exploration for depth map intra-frame prediction.

This paper presents a survey of 3D-HEVC depth map intra-frame prediction. It includes (i) a detailed description of depth map coding tools, (ii) an encoding time distribution analysis of these tools, and (iii) the block sizes and encoding mode distribution in the intra-frame prediction of depth maps. All evaluations were done considering the encoder implementation in the 3D-HEVC Test Model (3D-HTM) 16.0 [32]. The 3D-HTM timing analyses were done intending to show the relative increase in the computational effort required by the new tools for depth map coding. Correlating the encoding modes usage with the required computational effort allows supporting upcoming implementations and evaluating the relevance of each investigated tool better.

Several works focus on presenting the encoding tools of HEVC such as shown in [33], and the inter-frame and intra-frame predictions described in [34] and [35], respectively. Other works present overviews of 3D-HEVC such as [6], [36], and [37], displaying the 3D-HEVC and MultiView HEVC (MV-HEVC) standards. Besides, the work on [38] includes all standardized extensions of HEVC. The works [39] and [40] analyze the 3D-HEVC encoding time distribution focusing on the entire system and intra-frame prediction, respectively.

This paper extends [40] showing higher details of the 3D-HEVC depth map intra-frame prediction, through the inclusion of an encoding time distribution analysis, and inserting a mode distribution analysis. Besides, a depth analysis of the encoding time distribution and encoding modes distribution, together with the summarized explanation of intra-frame prediction tools for depth map coding, are the main contributions of this work regarding [6], [36], [37], [39] and [40].

Although depth map technology emerged with 3D-HEVC to fulfill 3D characteristics, this technology still will be used in future video coding standards and image processing applications such as Point Cloud [41] and light field coding [42]. Furthermore, designers can use the concepts and analysis discussed in this paper to plan new tools for complexity reduction, complexity control or even for encoding depth maps. As a practical example, JPEG-PLENO [43], which is still under standardization, has some proposals for using depth maps.

The remainder of this paper is divided as follows. Section II presents the depth map intra-frame prediction structure. Section III shows the new prediction modes applied in intra-frame prediction for depth map coding. Section IV presents the remaining encoding steps of depth map intra-frame prediction. Section V analyzes the depth map intra-frame prediction mode and the encoding time distribution of each encoding tool for intra-frame prediction, concerning all available block sizes. Finally, Section VI renders the main conclusions of this paper.

II. 3D-HEVC Depth Map Intra-Frame Prediction

The encoding structure of the 3D-HEVC depth maps follows the same quadtree principle of the HEVC texture coding [44]–[46]; i.e., the current frame should be divided into Coding Tree Units (CTUs) to be encoded, and the CTU sizes are 16 × 16, 32 × 32 and 64 × 64 pixels, where the 64 × 64 size is the most used. Next, CTU can be recursively split into four quadratic Coding Units (CUs) with sizes ranging from the CTU size to 8 × 8 pixels.

A CU can be further partitioned into Prediction Units (PUs), which assume the CU size or smaller quadratic, rectangular or asymmetric blocks, being capable of reaching a minimum size of 4 × 4 samples [47]. The number of available PU sizes depends on the used prediction type and CU size. There are four types of predictions used by 3D-HEVC: intra-frame, inter-frame, inter-component and inter-view [48]. The intra-frame prediction considers only previously processed samples of the current frame in the current prediction. The inter-frame prediction considers the information of previously processed frames in the prediction process [49]. The inter-component prediction uses the texture information to forecast the depth maps, and vice-versa [50]. Finally, the inter-view prediction considers the information of neighbor views in a MultiView scenario to make the prediction [51]. Therefore, the combination of these predictions can explore several redundancies present in a 3D video scenario.

Each block size can be encoded using several coding tools regarding the encoding block type (texture or depth map) and prediction type. The selected block size and coding tool are defined assessing the Rate-Distortion cost (RD-cost), which is a function that ponders the image quality and the number of bits need to encode the block with a given encoder configuration.

Fig. 2 illustrates the dataflow model of depth map intra-frame prediction for a given depth block, which is used in the 3D-HTM reference software [32]. The encoding configurations are evaluated according to its RD-cost, and the configuration with the lowest RD-cost is selected to encode the block.

3D-HEVC uses the same tool defined for texture intra-frame prediction in HEVC [35]. Both modes,

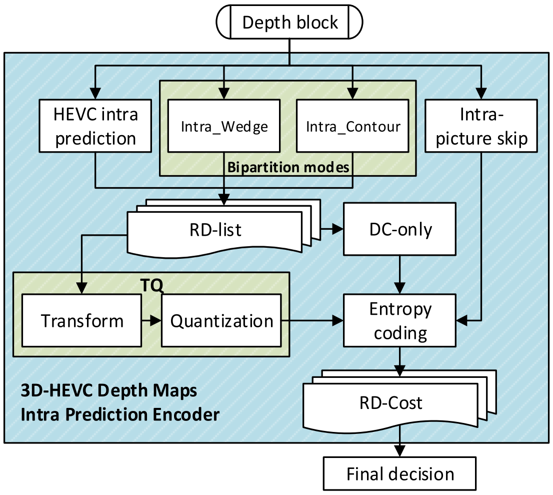


Fig. 2. Dataflow for intra-frame prediction of 3D-HEVC depth maps [40].

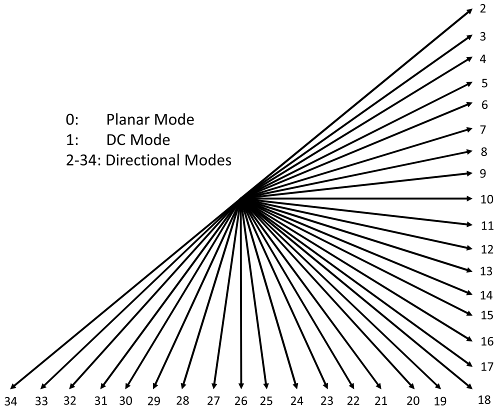


Fig. 3. HEVC intra-frame prediction modes [35].

Intra_Wedge and Intra_Contour [18] segment the encoding block into two regions, and each region is predicted with a constant containing the average of all values of the region. Intra_Wedge divides the encoding block using a straight line called wedgelet, while Intra_Contour creates a contour segmentation dynamically. The intra-picture skip defines four extra prediction modes for having high bitrate decrease in homogeneous regions [19].

The 3D-HEVC intra-frame prediction and the bipartition modes use a Rate-Distortion list (RD-list) to define some prediction modes that are further evaluated by their RD-cost, since evaluating all encoding possibilities by their RD-cost implies a prohibitive computation effort.

After processing the prediction tools, all modes inserted into the RD-list are evaluated using the Transforms followed by Quantization (TQ) and DC-only flows. Subsequently, the RD-costs are obtained using entropy coding in the results of TQ and DC-only flows. The TQ flow is the same applied in the HEVC texture coding, and the DC-only flow [20] has been designed as a new encoding alternative to the TQ flow, focused on obtaining higher efficiency in the depth maps of homogeneous regions. Additionally, the entropy encoder evaluates the intra-picture skip modes to obtain the RD-cost without using the RD-list.

The depth map intra-frame prediction tools are detailed in Section III, including the HEVC intra-prediction (Section III-A), bipartition modes (Section III-B) and intra-picture skip (Section III-C). The remaining encoding tools are described in Section IV, including TQ (Section IV-A), DC-only (Section IV-B) and entropy coding (Section IV-C).

III. Prediction Tools

This section presents the depth map intra-frame prediction tools, including Intra_Wedge, Intra_Contour, and intra-picture skip, as well as the conventional HEVC intra-prediction.

A. HEVC Intra-Frame Prediction

The 3D-HEVC depth map intra-frame prediction uses the same algorithm defined in HEVC for texture coding. It defines the planar mode, the Direct Component (DC) mode, and 33 directional modes (i.e., 35 prediction modes), whose directions are presented in Fig. 3 [35]. These modes use samples of neighboring blocks as references for creating a predicted block. The intra-frame prediction tools apply a specific number of directions according to the available block sizes; i.e., (i) 18 directions for 4 × 4 blocks; (ii) 35 directions for 8 × 8, 16 × 16 and 32 × 32 blocks; and (iii) 4 directions for 64 × 64 blocks.

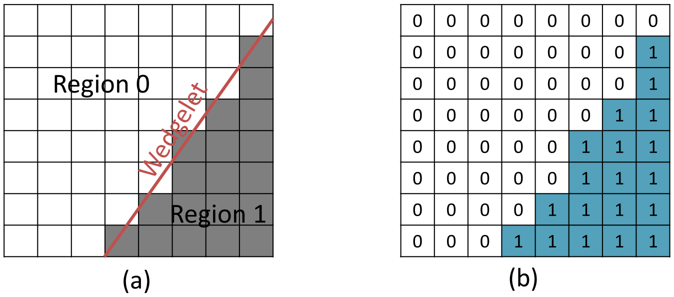
The exhaustive encoding approach, which evaluates the RD-cost of all available prediction modes, is prohibitive for real-time applications. Thus, 3D-HEVC Test Model software (3D-HTM) employs the heuristic proposed by Zhao *et al.* [52] to encode texture and depth maps, generating an RD-list with few modes among all available ones, and the RD-cost is computed only for these selected modes.

The heuristic proposed by Zhao *et al.* [52] uses Rough Mode Decision (RMD) and Most Probable Modes (MPM) algorithms. RMD uses Sum of Absolute Transformed Differences (SATD) between the original block and the predicted one to evaluate early the intra modes (without the complete RD-cost evaluation). The algorithm orders the modes according to their SATDs and inserts the modes with the lowest SATDs ordered into the RD-list (8 modes for 4 × 4 and 8 × 8 blocks, and 3 modes for 16 × 16, 32 × 32, and 64 × 64 blocks). Subsequently, the MPM algorithm gets the used modes in the encoded neighbor blocks (the left and above neighbors) and inserts them into the RD-list.

B. Bipartition Modes

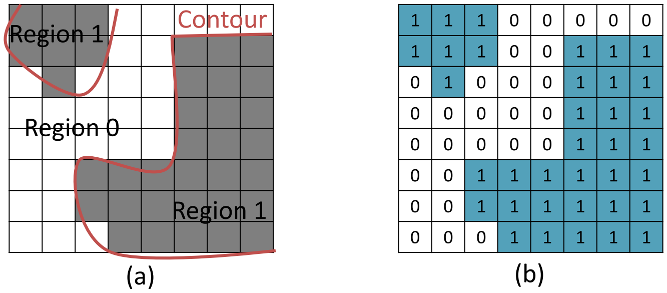
The bipartition modes are a novelty of the 3D-HEVC depth map intra-frame prediction. They include the Intra_Wedge and Intra_Contour modes, which produce wedgelet (Fig. 4) and contour segmentations (Fig. 5) for block sizes from 4 × 4 to 32 × 32. These modes are not available for 64 × 64 blocks.

Fig. 4(a) shows the wedgelet segmentation pattern divides the block with a straight line and Fig. 4(b) illustrates a discretization performed to identify the region to which the samples near the straight line belong which are not entirely within a single region.



0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1
0	0	0	0	0	1	1	1
0	0	0	0	0	1	1	1
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	1

Fig. 4. Wedgelet segmentation model of a depth block: (a) pattern with Regions 0 and 1 and (b) discretization with constant values [30].



1	1	1	0	0	0	0	0
1	1	1	0	0	1	1	1
0	1	0	0	0	1	1	1
0	0	0	0	0	1	1	1
0	0	0	0	0	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1

Fig. 5. Contour segmentation model of a depth block: (a) pattern with Region 0 and Region 1 and (b) discretization with constant values [30].

TABLE I

NUMBER OF EVALUATED WEDGELETS IN INTRA_WEDGE

Block size	Total of possible wedgelets	Evaluated wedgelets in the main stage
4×4	86	58
8×8	802	314
>= 16×16	510	384

While Intra_Wedge mode assumes only pre-defined patterns, Intra_Contour mode employs a contour segmentation for modeling arbitrary patterns consisting of one or more parts (but only two regions with constant values are allowed). Fig. 5(a) exemplifies the contour segmenting of a block into two regions, and Fig. 5(b) presents the contour discretization with the constant value of each region. Details of the Intra_Wedge and Intra_Contour modes algorithms are presented next.

1) Intra_Wedge Mode: The Intra_Wedge mode defines the evaluation of several wedgelets for each depth map block; however, only a subset of the complete wedgelet set should be evaluated to reduce the Intra_Wedge encoding time. Table I displays the number of possible wedgelets in a depth map block according to the size of this block.

Fig. 6 presents a high-level diagram of the Intra_Wedge algorithm, which is composed of three stages: (i) Main Stage, (ii) Refinement Stage, and (iii) Residue Stage.

The Main Stage evaluates the initial wedgelet set (i.e., wedgelets assessed before the refinement) and finds the best wedgelet partition among the available ones. The process of finding the best wedgelet requires mapping the encoding block into the binary pattern defined by each wedgelet. According to this mapping, the average values of all samples mapped into regions 0 and 1 are computed, and the predicted block is defined as the average value of each region (Prediction step). Next, for each wedgelet pattern, a similarity criterion is applied to determine the encoding efficiency of this pattern in comparison to the other wedgelet patterns; this procedure generates a local distortion value. All local distortions are compared, and the pattern with the lowest distortion is selected as the best wedgelet (Distortion step). There are many possibilities for similarity criterion usage, such as Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD), and Synthesized View Distortion Change (SVDC) [54]. Each similarity criterion should be capable of obtaining different impact and encoding time profiling. 3D-HTM 16.0 employs the SVDC algorithm as default; it compares the synthesized view generated using the evaluated pattern to the one made with the original depth samples to evaluate the encoding distortion.

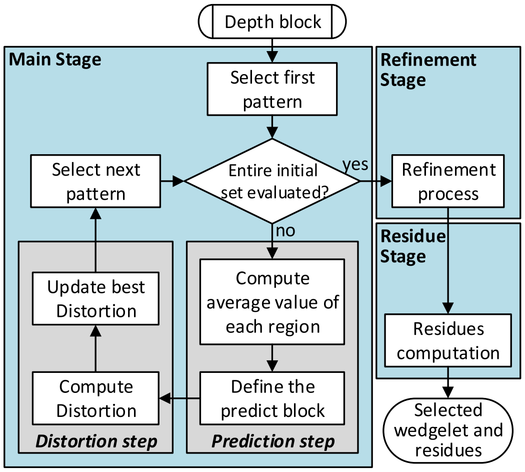


Fig. 6. Main modules of the Intra_Wedge encoding algorithm [53].

The Refinement Stage evaluates up to eight wedgelets around of the selected one (i.e., with a similar pattern) in the previous operation. Again, the wedgelet with the lowest distortion among these eight possibilities, along with the wedgelet selected in the Main Stage, is selected as the best one.

Finally, the Residue Stage subtracts the predicted block from the original one and adds this Intra_Wedge prediction into the RD-list.

2) Intra_Contour Mode: The Intra_Contour mode uses the inter-component prediction technique to find the best contour partition. This prediction applies the previously encoded information from texture during the prediction of depth maps [36].

A motivation for the Intra_Contour design is the depth maps represent the same viewpoint of the associated texture view, which was previously encoded. Depth maps and texture contain different characteristics; however, both exhibit structural similarities that can be explored. For instance, an edge in the depth component usually corresponds to an edge in the texture component; thus, there is significant redundant information between both. Fig. 7 highlights this redundancy showing the high structural similarity and correlation of both blocks.

Fig. 8 presents the Intra_Contour encoding flow. The only information known at the beginning of the Intra_Contour

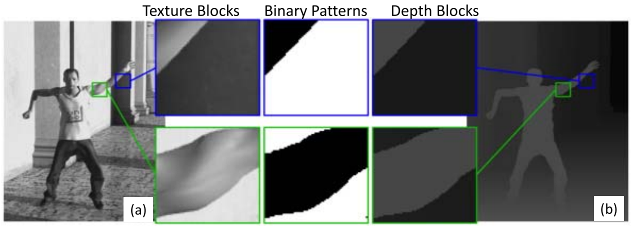


Fig. 7. Correlation between (a) texture and (b) depth components [55].

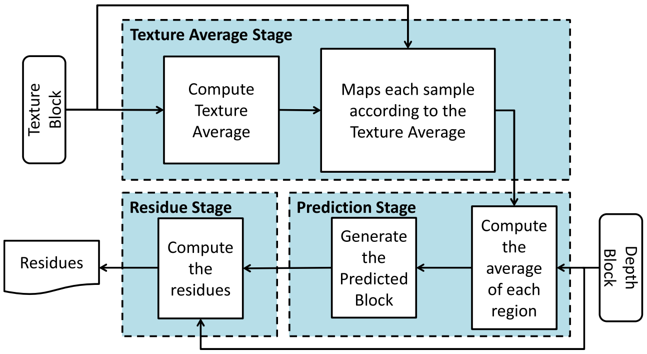


Fig. 8. Intra_Contour encoding flow.

```
for i = 1 to N
    for j = 1 to N
        if sample[i][j] < u
            Region 0 ← sample[i][j]
        else
            Region 1 ← sample[i][j]
```

Fig. 9. Pseudo-code of part of the Intra_Contour mode representing the mapping of samples into Region 0 and Region 1.

execution is the texture and depth blocks. Besides, only the reconstructed luminance signal (previously encoded) composes the texture block used as a reference in the Intra_Contour.

Three stages compose the Intra_Contour encoding algorithm: Texture Average Stage, Prediction Stage, and Residue Stage.

The Texture Average Stage starts computing the average value (u) of the corners samples in the texture block. Fig. 9 shows a pseudo-code for creating a binary map that determines the Intra_Contour partition, where N is the block width. The pseudo-code generates a binary map without the use of depth information by mapping into Regions 0 or 1 all texture samples smaller or greater than u, respectively. Since the texture block is decoded before the depth block, the Intra_Contour partition does not need to be transmitted, being generated adaptively at the decoder side, without bitrate increase.

The Prediction Stage processes the depth map using the binary map determined in the previous stage and calculates the average values of all depth block samples mapped into Regions 0 and 1. These values are selected as the predicted block, according to the binary map value.

Finally, the Residue Stage of Intra_Contour subtracts the predicted block from the original depth block, generating the residues and inserting this mode into the RD-list.

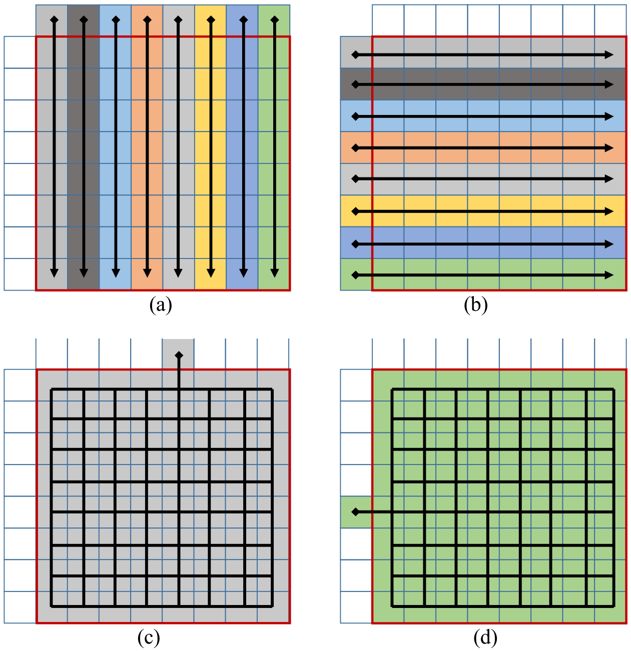


Fig. 10. The four intra-picture skip encoding modes for an 8 × 8 encoding block. (a) Mode 0 - the above samples are copied vertically to the predicted block. (b) Mode 1 – the left samples are copied horizontally to the predicted block. (c) Mode 2 - the middle-upper sample is copied to the entire predicted block. (d) Mode 3 - the middle-left sample is copied to the entire predicted block.

C. Intra-Picture Skip

The intra-picture skip is another novelty of the 3D-HEVC depth map intra-frame prediction, which was developed considering most of the depth maps are smooth areas for block sizes ranging from 8 × 8 to 64 × 64. In general, slight changes of depth values in flat areas are not relevant to synthesized views quality [19]. Thus, intra-picture skip avoids the residual coding for the smooth areas providing a significant bitrate reduction.

The intra-picture skip includes four additional prediction modes [19]: (i) Mode 0 (vertical intra-frame), which copies all neighbor upper samples to the current encoding block; (ii) Mode 1 (horizontal intra-frame), which copies all neighbor left samples to the entire current encoding block; (iii) Mode 2 (single vertical depth), which copies only the middle upper neighbor sample to the entire current encoding block and; (iv) Mode 3 (single horizontal depth), which copies only the middle left neighbor sample to the entire current encoding block.

Fig. 10(a) to (d) exemplify the intra-picture skip modes 0, 1, 2 and 3, respectively, regarding an 8 × 8 encoding block. Only the N above and N left neighbor samples are used in the prediction (considering N^2 the number of samples of the square encoding block). The HEVC vertical (Fig. 3, direction 26) and horizontal (Fig. 3, direction 10) intra directions are used to generate the first two modes (i.e., Modes 0 and 1). The vertical and horizontal intra single depth modes (i.e., Modes 2 and 3) fill the predicted block with a single depth value derived from the upper and left neighbor blocks, respectively. In this case, it is not necessary to transmit the encoding block nor extra information. Only side information is required to be entropy coded and packed in the bitstream, providing significant bitrate reduction. Hence, its RD-cost computation is mainly composed of its distortion computation.

IV. TQ, DC-Only and Entropy Coding

After the execution of the prediction steps, the depth maps encoding is finalized by applying three additional tools: Transform-Quantization (TQ), DC-only, and entropy coding. The modes inside the RD-list, generated by standard HEVC intra-frame prediction and by the bipartition modes, are processed by TQ or DC-only together with entropy coding. Since the residues are not necessary to evaluate the intra-picture skip modes, its results are processed directly by entropy coding. Next subsections describe these tools.

A. Transform-Quantization (TQ)

The TQ flow of the 3D-HEVC depth maps uses the same process applied in the HEVC texture coding; it divides the CU in a quadtree of Transform Unities (TUs) [56], which allows TQ varying from 4 × 4 to 32 × 32 samples. Discrete Cosine Transform (DCT) and a quantization module process the residual information of the predicted block under evaluation attenuating or removing the less relevant frequencies to the human vision [57]; thus, generating quality losses in the coding process. A Quantization Parameter (QP) defines the quantization granularity; i.e., the larger the QP value, the higher the compression rate and image degradation [57].

B. DC-Only

DC-only is another innovation of the 3D-HEVC intra-frame prediction used for block sizes ranging from 8 × 8 to 64 × 64. It is an alternative method to the transform-based coding for getting further bitrate gains in smooth regions [20]. Instead of using the residual data of depth luminance samples, DC-only considers a single DC residue to codify the HEVC intra-frame prediction modes and two DC residues to codify the bipartition modes.

The DC residue is computed by subtracting the original DC ($\mathrm{dc_o}$) from a predicted DC ($\mathrm{dc_p}$). The $\mathrm{dc_o}$ value is calculated using the average value of the original block for HEVC intra-frame prediction and the average value of each region for the case of bipartition modes. The $\mathrm{dc_p}$ value is generated using the average value of the four corners samples of the predicted block for intra-frame prediction, whereas $\mathrm{dc_p}$ uses the predicted Constant Partition Value (CPV) obtained by applying the procedure defined in [58] for bipartition modes.

C. Entropy Coding

The entropy coding employs the same tools defined in the HEVC encoding process. The main tool used is the Context Adaptive Binary Arithmetic Coding (CABAC) [59], which can provide a significant bitstream reduction by performing lossless entropy compression in the series of syntax elements delivered by the previous tools. Thus, all information generated by the previous encoding tools (TQ, DC-only, and intra-picture skip) are processed by the entropy coding. The entropy coding results allow calculating the RD-cost of all possible modes, enabling to choose the one used to encode the current image block. The entropy code of the selected mode (together with side information) is inserted into the bitstream.

V. Depth Map Intra-Frame Prediction Analysis

Subsection V-A presents the encoding time distribution, and Subsection V-B shows the encoding modes and blocks size usage in the depth map coding. All the analysis about the depth map intra-frame prediction follows the latest Common Test Conditions (CTC) for 3D videos experiments described in [60].

The CTC experiments for 3D videos have been developed to be a benchmark to evaluate coding tools. The videos need to be encoded inside CTC contain several specific characteristics. However, CTC is being regularly enhanced with the insertion of new videos and/or different coding settings; therefore, CTC is related to a specific time-period. The latest CTC requires the encoding of eight pre-defined 3D videos with distinct characteristics in four quantization scenarios. Three videos inside CTC have 1024 × 768 resolution (Balloons [11], Kendo [11] and, Newspaper [61]) and the other five have 1920 × 1088 resolution (GT_Fly [62], PoznanHall2 [63], PoznanStreet [63], UndoDancer [64] and, Shark [65]). The quantization scenarios are divided into four pairs of quantization parameters (QP-pair - QP-texture/QP-depth), where their values are (25/34), (30/39), (35/42), and (40/45). As explained before, the higher is the QP-pair, the higher is the reached compression rate and the image quality degradation. QP-texture is the part of QP-pair used in the texture coding, while QP-depth is the part of QP-pair relative to the QP value used to encode the depth maps.

The experiments presented in this paper consider all videos and QP-pairs defined by the CTC in [60] and, using the all intra-frame configuration. The version 16.0 of the 3D-HTM [32] was used to generate the evaluation results. The algorithms employed in these evaluations already include bipartition modes and DC-only timesaving techniques introduced by the works [24] and [26], respectively, since 3D-HTM 16.0 integrated them by default.

A. Encoding Time Distribution

All experiments described in this paper were conducted in a server encompassing two Xeon E5-2660 processors (10 cores and 20 threads) with 96 GB of main memory.

Table II presents the results of the first experiment considering the percentage of the encoding time distribution for each component per video and per QP-pair. The average time for encoding one texture frame and its associated depth map is also presented in Table II. The encoding time varies according to the video characteristics; however, on average, the encoding time decreases when the QP increases.

The encoding time spent during texture coding ranges from 11.0% to 18.2%, which is much lower than depth map

TABLE II
ENCODING TIME EVALUATION FOR TEXTURE AND DEPTH CODING IN THE 3D-HEVC ALL INTRA CASE

Videos	QP-pair											
	(25/34)			(30/39)			(35/42)			(40/45)		
	Texture (%)	Depth (%)	Enc. time (s)	Texture (%)	Depth (%)	Enc. time (s)	Texture (%)	Depth (%)	Enc. time (s)	Texture (%)	Depth (%)	Enc. time (s)
Balloons	15.7	84.3	33.72	14.4	85.6	34.78	14.9	85.1	33.64	14.5	85.5	34.48
Kendo	15.9	84.1	31.62	15.2	84.8	32.90	15.4	84.6	32.52	13.4	86.6	32.77
Newspaper	13.2	86.8	45.62	11.0	89.0	45.53	11.7	88.3	42.66	11.8	88.2	42.21
GT_Fly	14.8	85.2	83.11	13.3	86.7	82.58	14.5	85.5	70.58	14.8	85.2	67.04
PoznanHall2	18.2	81.8	47.67	16.2	83.8	49.60	16.8	83.2	47.72	15.6	84.4	49.20
PoznanStreet	13.6	86.4	95.89	12.2	87.8	93.01	13.7	86.3	77.32	13.5	86.5	73.96
UndoDancer	18.0	82.0	78.98	16.7	83.3	73.53	16.0	84.0	68.47	14.8	85.2	67.37
Shark	16.1	83.9	87.93	14.7	85.3	88.99	15.2	84.8	80.80	14.5	85.5	80.22
Average	**15.7**	**84.3**	**63.07**	**14.2**	**85.8**	**62.62**	**14.8**	**85.2**	**56.71**	**14.1**	**85.9**	**52.43**

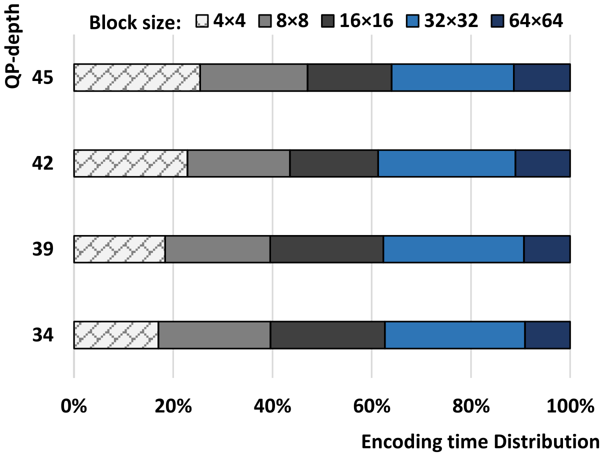


Fig. 11. Encoding time distribution according to the block size and QP-depth.

encoding time. It occurs in all intra-frame scenarios because texture coding only applies traditional HEVC intra-frame prediction following the TQ flow, while depth coding still uses Intra_Wedge, Intra_Contour, intra-picture skip, and DC-Only evaluations. On average, depth map coding is 5.8 times more complex than the texture coding in the all intra-frame scenario.

Two other encoding time evaluations complement this experiment. The first one describes the percentage of encoding time used per block size and the second experiment describes the encoding time behavior inside a given block size.

Fig. 11 illustrates the distribution evaluation of the block size encoding time according to the used QP-depth. The encoding time axis represents the percentage of time required for depth map encoding for a given scenario.

The QP-depth variation produces little different encoding time distribution because QP-based timesaving algorithms [26] and [24] are being applied. However, it is expected the use of different block sizes varies according to the QP-depth, as discussed in Subsection V-B. Thus, this fact can bring several benefits when designing a complexity reduction solution. Besides, it is vital to identify the time spent on the encoding tools. For instance, some modes share encoding steps (e.g., the RMD and MPM selection in HEVC intra-frame prediction are used for both TQ and DC-only flows); therefore, the next analysis regards the encoding steps instead of the encoding tools. The following steps were considered for this encoding time distribution evaluation: (i) RMD and MPM selection; (ii) Intra_Wedge wedgelet search; (iii) Intra_Contour pattern generation; (iv) RD-list evaluation in the TQ flow; (v) RD-list assessment in the DC-only flow; and (vi) intra-picture skip evaluation. Steps (iv) to (vi) also consider the time spent in entropy coding to generate the RD-cost.

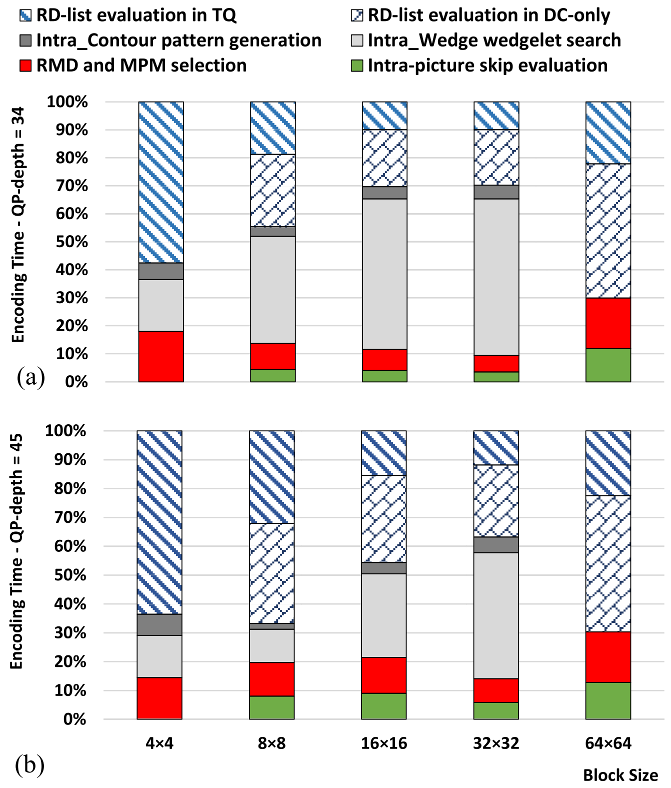


Fig. 12. Encoding time distribution according to the block size - (a) QP-depth = 34 and (b) QP-depth = 45.

Fig. 12 shows the encoding time distribution among the intra-frame prediction steps for each block size, regarding the corners QP-depth values, which complements the previous experiment.

Firstly, one can notice the 4 × 4 blocks do not present intra-picture skip and DC-only evaluations because these modes are applied only in the block sizes ranging from 8 × 8 to 64 × 64. Besides, Intra_Wedge and Intra_Contour are not present in 64 × 64 blocks because bipartition modes are only applied in the block sizes ranging from 4 × 4 to 32 × 32.

Regarding 4 × 4 blocks, where intra-picture skip and DC-only evaluations are not available, the RD-list evaluation in the TQ flow is the most time-demanding operation for both evaluated QP-depths. Intra_Wedge presented the second highest percentage of encoding time. The RMD and MPM selection and Intra_Contour pattern generation also use a significant amount of time to be processed, when compared to the other block sizes.

The 8 × 8 block size presents a significant variation in the time distribution of the encoding tools when QP-depth changes. For lower QP-depths, the Intra_Wedge wedgelet search represents the highest percentage of time among the encoding steps. This behavior changes entirely for high QP-depths, where the RD-list evaluation in DC-only and TQ have the highest percentage of time demanded. This behavior occurs because the bipartition modes were proposed to handle with the high detail levels of depth maps, intending to preserve the borders, as previously discussed. Higher QP-depths values reduce the encoded image details; therefore, the bipartition modes tend to be fewer selected, increasing the percentage of use of the remaining encoding tools. Therefore, applying the heuristic proposed in [26] by default in 3D-HTM, the higher QP-depth values tend to accelerate the Intra_Wedge execution.

The Intra_Wedge wedgelets search and the RD-list evaluation in DC-only present the first and second positions, respectively, in the percentage of time spent to encode 16 × 16 and 32 × 32 block sizes for the evaluated QP-depths. The Intra_Wedge execution spent a higher percentage of time for higher block sizes because the solution proposed in [26], which is used in 3D-HTM, removes the bipartition modes evaluation for low variance blocks. Since larger blocks tend to have higher variance, then fewer skips of bipartition modes should occur.

When 64 × 64 blocks are processed, and bipartition modes are not available, the RD-list evaluation in DC-only is the most time-consuming operation, representing almost 50% of the total encoding time for both QP-depths. The DC-only execution time increases with the block size increase because DC-only was planned to be used in blocks composed of single homogeneous areas or blocks that can be partitioned into two homogeneous areas. Thus, the heuristic proposed by [24] tends to skip the DC-only evaluation for heterogeneous regions.

Intra_Contour and intra-picture skip predictions modes have slight execution times for almost all block sizes. Besides, for the 64 × 64 encoding blocks, the execution time of Intra_Contour is insignificant, and this is the only block size that intra-picture skip reaches more than 10% of the execution time.

Meaningful conclusions can also be obtained about the time distribution of encoding steps according to the QP-depth values. For example, the percentage of time spent in the Intra_Wedge calculation is inversely proportional to the QP-depth values used for all block sizes. This occurs because the heuristic proposed in [26] reduces bipartition modes evaluations when high QP-depth values are used. Consequently, lower QP-depths imply lower skips on bipartition modes. The opposite behavior occurs for the RD-list evaluation in DC-only, which is higher for all block sizes using high QP-depths. It happens because if the encoding block is not homogeneous, the DC-only evaluation is skipped [24] and higher QP-depth values tend to generate more homogeneous areas, which are encoded efficiently by DC-only. The RD-list evaluation in TQ also presents a similar behavior as the RD-list evaluation in DC-only, where higher QP-depth values raise the percentage of time spent in the RD-list evaluation.

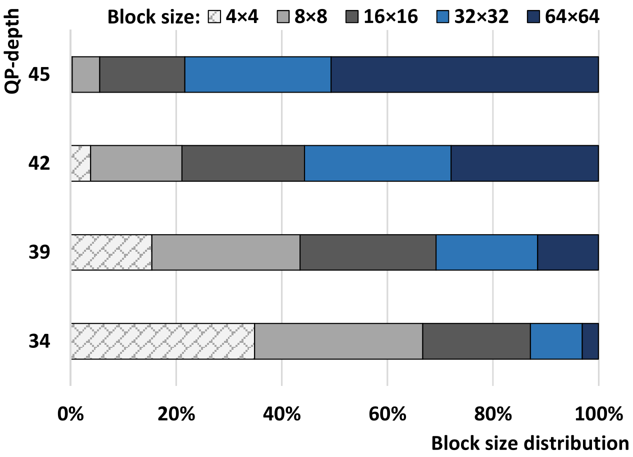


Fig. 13. Block size distribution for the different quantization scenarios.

The intra-picture skip evaluation, RMD and MPM selection and Intra_Contour pattern generation do not present a significant encoding time variation with the QP-depth change.

B. Encoding Modes and Block Size Usage Distribution

Fig. 13 shows the percentage of block sizes used to encode the depth maps according to the QP-depths. The experiment regards average results for all videos inside CTC when assessed under all intra-frame configuration.

The selection of block size depends on the QP-depth value; high QP-depth values imply selecting bigger blocks (e.g., 4 × 4 and 8 × 8 block sizes are used more than 65% of times for QP-depth = 34), and the opposite is also true (e.g., 32 × 32 and 64 × 64 block sizes are used more than 75% of times for QP-depth = 45). Besides, the percentage of 16 × 16 blocks selection remains nearly constant, independently of the QP-depth. This happens because low QP-depths tend to preserve the depth maps details, generating heterogeneous regions, which are encoded better with smaller block sizes. On the other hand, high QP-depths tend to attenuate the details of the depth maps, creating homogeneous regions, which are encoded better with bigger block sizes. Besides, it is essential to understand the encoder selection among the available encoding modes for having a better understanding of the encoder decisions. This analysis is presented next, considering five possibilities of encoding flows: (i) *Bp-TQ* – a bipartition mode encodes the block using the TQ flow; (ii) *Intra-TQ* – the HEVC intra-frame prediction encodes the block using the TQ flow; (iii) *Bp-DC* – a bipartition mode encodes the block applying the DC-only flow; (iv) *Intra-DC* – the HEVC intra-frame prediction encodes the

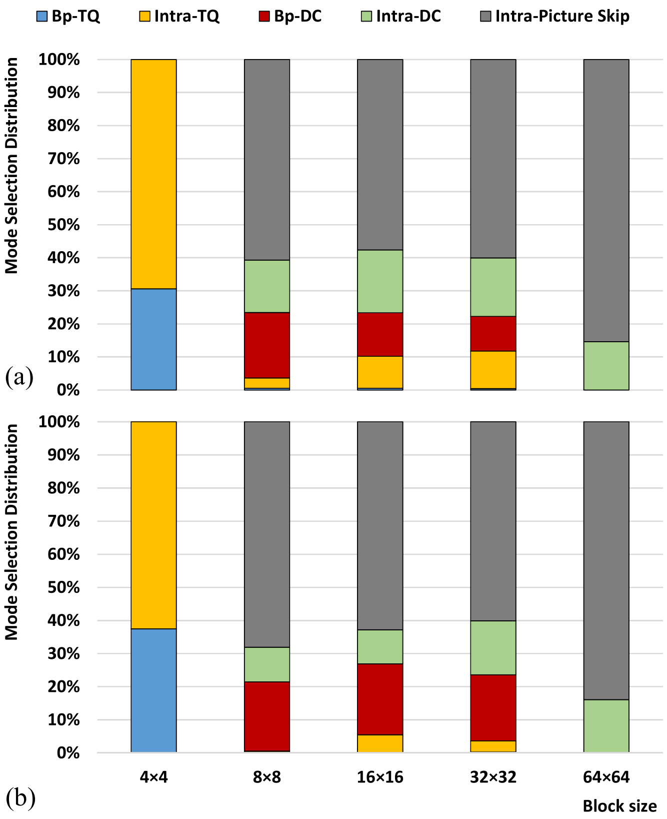


Fig. 14. Average mode distribution for each block size with (a) QP-depth = 34 and (b) QP-depth = 45.

block applying the DC-only flow, and; (v) *Intra-picture skip* – any intra-picture skip mode encodes the block.

Given the two corners of QP-depths in CTC, Fig. 14 shows the mode distribution selection for each block size has a low level of dependency with the QP-depth value.

Regarding 4 × 4 blocks, on average, Intra- and Bp-TQ flows were chosen 68% and 32% of times, respectively. This is the unique block size where the Intra- or Bp-TQ flows are employed in a relevant number of times since intra-picture skip and DC-only are not available.

All the five flows were evaluated concerning blocks ranging from 8 × 8 to 32 × 32. In this range of blocks, independently of the QP-depth, the intra-picture skip encoding flow was selected more than 50% of times. The DC-only evaluation through Bp-DC and Intra-DC are also significant in this distribution since both encoded flows are selected 33%, on average. Intra-TQ and Bp-TQ are the less representative encoding flows. Intra-TQ is selected a maximum of 11% (QP-depth = 34 and 32 × 32 block size) and a minimum of less than 1% (QP-depth = 45 and 8 × 8 block size). Bp-TQ was selected less than 1% for all scenarios. For these intermediate block sizes, the higher the QP, the higher the use of intra-picture skip and Bp-DC, and the smaller the use of Intra-DC, Intra-TQ, and Bp-TQ. Besides, Bp-TQ usage tends to zero for the highest value of QP-depth.

Only intra-picture skip, Intra-DC, and Intra-TQ modes are evaluated with the 64 × 64 blocks. However, the experimental results show Intra-TQ is almost not used. The intra-picture skip mode is selected in more than 85% of times, followed by Intra-DC. It happens mainly because larger encoding blocks are chosen when a region with low details is encoded such as the background or the body of objects with depth maps composed of large homogeneous areas. In this case, Intra-TQ is almost irrelevant since intra-picture skip and Intra-DC can reduce the bitrate expressively without compromising the visual quality.

Employing the same colors of Fig. 14, Fig. 15 exemplifies the five encoding flows of the central view of the first frame of two videos available in CTC (Balloons [11] and Shark [65]). The box containing the color of the used mode was plotted in front of the original depth image using 50% of transparency. The frames encoded using the lowest and the highest QP-depth values show that higher values of QP-depth significantly reduces the image details, generating smooth regions. On the other hand, this means the efficient encoding of depth maps with high QP-depths requires bigger block sizes. On the other hand, small block sizes are suitable to encode the details of depth maps using low QP-depths.

Additionally, Fig. 15 shows Bp-TQ, Bp-DC, and Intra-TQ are used mainly in the borders of the objects, while intra-picture skip and Intra-DC are selected in regions with smooth areas such as bodies and backgrounds. Intra-DC can also achieve a high usage in blocks that can be partitioned into two homogeneous regions. As the QP-depth increases, the quantity of smooth area also increases, raising the usage of intra-picture skip and Intra-DC.

It is crucial to identify the high use of intra-picture skip encoding mode concentrated in the background and the bodies of objects. Intra-picture skip is hardly used in the borders of the objects since it does not encode residual information. However, intra-picture skip obtains good results on the object borders, when a horizontal or a vertical line segments the block. In these cases, the intra-picture skip Modes 0 or 1 should be selected, as can be seen in Balloons video sequence.

Remarks the information about the decoding mode distribution of the encoded blocks in the stream allows understanding and estimating the decoding times of this stream since all blocks of the stream are decoded with the modes selected during the coding stage. We emphasize the Intra_Contour tool, of the intra-frame prediction decoder, requires extra computational effort since it needs to build all the patterns dynamically to reconstruct the blocks. Besides, the computational effort to decode blocks of different sizes grows with the number of pixels of each block.

C. Timesaving Possibilities

Considering the analyses discussed in previous sections, many conclusions can be taken intending to reduce the encoding time of the depth map intra-prediction, while maintaining the encoding efficiency. Nevertheless, it is important to highlight all new depth map intra-coding tools were added to the 3D-HEVC specification because they enhance the encoding efficiency significantly. More precisely, bipartition modes can better represent an edge region, enhancing the quality of the synthesized views considerably, where the depth map borders

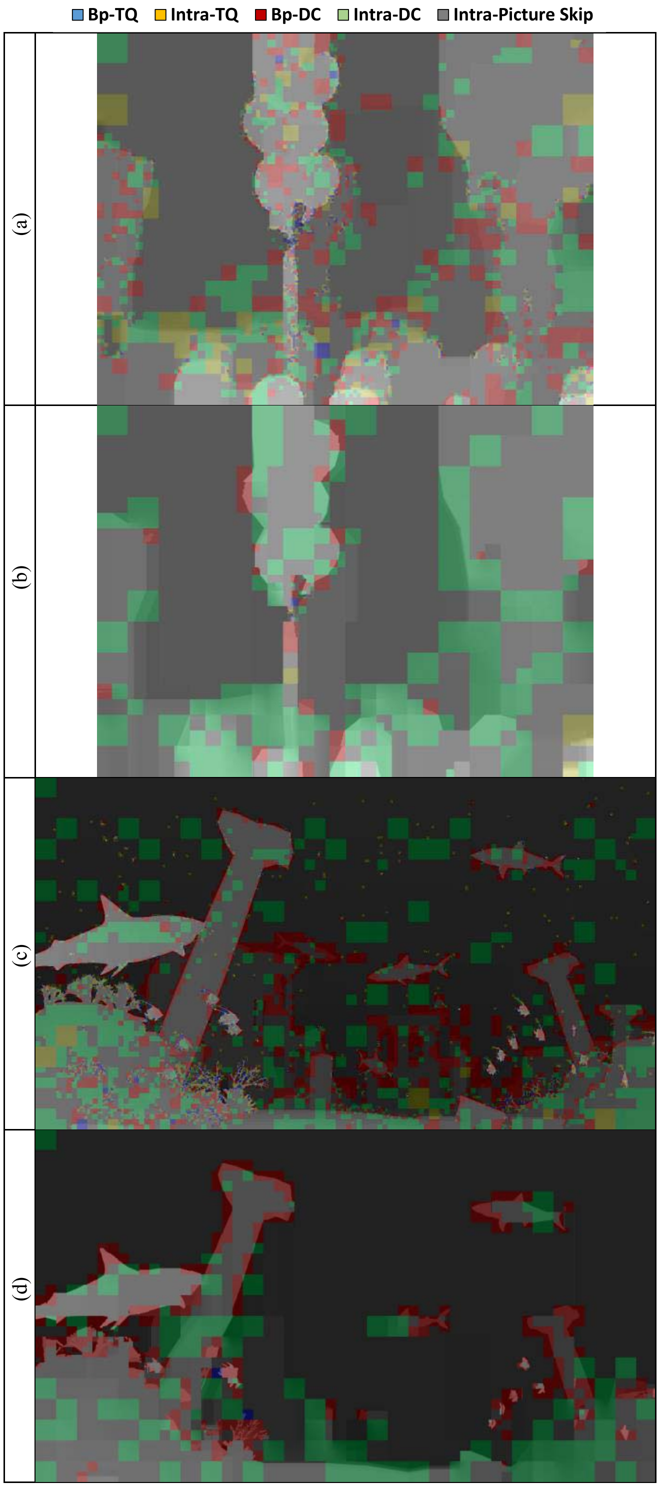


Fig. 15. Examples of mode distribution of the central view of the first frame of Balloons (a) QP-depth = 34, (b) QP-depth = 45 and Shark (c) QP-depth = 34 and (d) QP-depth = 45 videos.

are critical. Intra-picture skip and DC-only reduce the required bitrate meaningfully for frame regions that are not critical for the view synthesis process. Therefore, these encoding tools cannot be merely removed to save time; otherwise, the encoding efficiency provided by 3D-HEVC concerning previous standards is drastically reduced. Nevertheless, some aspects described next can be explored to reduce the encoding time without significant loss on the encoding efficiency. Besides, it is relevant to mention the time-saving algorithms in intra-frame prediction provide little efficiency with Random Access (RA) configuration because the execution of various depth-coding tools is skipped in P- or B- frames; so, the RA configuration is not exploited in this work.

First, encoding small blocks such as 4 × 4 and 8 × 8 takes more time compared to larger blocks, independent of the QP-depth. Moreover, the selection of small blocks is reduced with the QP-depth grows. The inverse holds true since larger blocks are hardly used with low QP-depth values. However, the encoding effort spent in their execution holds similar among different evaluated QP-depths. Therefore, the QP-depth value can be considered when designing a timesaving solution by limiting the depth and the top level of the quadtrees.

Another valuable concern is the TQ flow is almost not used for blocks larger than 4 × 4, and the higher the QP-depth, the lower the use of TQ. However, when analyzing the encoding effort spent in its evaluation, it keeps a significant percentage of encoding time. Then, solutions that reduce RD-list assessment in TQ flow for larger block sizes are relevant and should consider the encoding QP-depth value to provide higher evaluations skips.

The previously proposed ideas are interesting to be performed when analyzing the mode distribution. However, it is also possible to reduce the computation time by simplifying the algorithms with longer encoding time. For 4 × 4 blocks, a local timesaving technique should consider reducing the number of modes inserted into the RD-list. The TQ flow should evaluate all modes inside this list, and then reducing the number of modes tends to save much time for 4 × 4 blocks.

When considering 8 × 8 to 32 × 32 blocks and lower QP-depths, the timesaving techniques should target the Intra_Wedge to reach more impressive results. When using high QP-depths, the algorithm proposed in [26] uses a high threshold resulting in more bipartition mode skips. Thus, Intra_Wedge encoding time is reduced a lot in this scenario, and additional timesaving solutions should not achieve remarkable results for Intra_Wedge. In this case, the simplification focus should be on other tools such as the TQ and DC-only flows evaluation. For instance, the simplification focus with 64 × 64 blocks should be the DC-only technique since its evaluation represents almost 50% of the encoding time.

The RMD and MPM processes are responsible for a considerable amount of encoding time to select the candidates that should be inserted into the RD-list for all available sizes of blocks. These algorithms were developed for texture coding, and the depth coding inherited them without modifications. Then, simplifying the depth map characteristics saves the overall encoding time and few works proposed solutions in this direction. Moreover, if the RMD and MPM selections reduce the number of modes inserted into the RD-list, then the encoding time of the TQ and SDC flows also decreases since they should evaluate a low amount of modes.

VI. Conclusions

This work presents a survey of intra-frame prediction in 3D-HEVC depth maps. This type of prediction implies much more encoding time than the texture intra-frame prediction since it encompasses new encoding tools such as bipartition modes, DC-only, and intra-picture skip; besides it also uses the HEVC intra-frame prediction and the Transform-Quantization flow. This work describes the most relevant tools required by intra-frame prediction and discusses their usage and encoding time distribution in the quantization scenarios defined by Common Test Conditions. This paper is the first one proposing these analyses, aiming to support future works on depth map coding for designing specific solutions. These solutions could include complexity reduction, complexity control, real-time embedded system implementation, and even the creation of new tools for depth map coding for future video coding applications.

ACKNOWLEDGMENT

This article was achieved in cooperation with Hewlett-Packard Brazil Ltda. using incentives of Brazilian Informatics Law (Law n° 8.248 of 1991).

REFERENCES

[1] C. Fehn, E. Cooke, O. Schreer, and P. Kauff, "3D analysis and image-based rendering for immersive TV applications," *Signal Process., Image Commun.*, vol. 17, no. 9, pp. 705–715, Oct. 2002.

[2] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel, "Free-viewpoint video of human actors," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 569–577, 2003.

[3] M. Grellert, S. Bampi, and B. Zatt, "Complexity-scalable HEVC encoding," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2016, pp. 1–5.

[4] Z. Liu *et al.*, "H.264/MVC interleaving for real-time multiview video streaming," *J. Real-Time Image Process.*, vol. 10, no. 3, pp. 501–511, Sep. 2015.

[5] JCT-3V. *JCT-3V Document Management System*. Accessed: Sep. 2017. [Online]. Available: http://phenix.int-evry.fr/jct2

[6] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, Jan. 2016.

[7] P. Kauff *et al.*, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process., Image Commun.*, vol. 22, no. 2, pp. 217–234, Feb. 2007.

[8] C. Bal and T. Q. Nguyen, "Multiview video plus depth coding with depth-based prediction mode," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 995–1005, Jun. 2014.

[9] L. Zhang and W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV," *IEEE Trans. Broadcast.*, vol. 51, no. 2, pp. 191–199, Jun. 2005.

[10] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," *Proc. SPIE*, vol. 5291, pp. 93–104, May 2004.

[11] Tanimoto Lab. *Nagoya University Multi-view Sequences Download List*. Accessed: Sep. 2017. [Online]. Available: http://www.tanimoto.nuee.nagoya-u.ac.jp

[12] P. Merkle, K. Müller, and T. Wiegand, "Coding of depth signals for 3D video using wedgelet block segmentation with residual adaptation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2013, pp. 1–6.

[13] K. Müller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proc. IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.

[14] P. Merkle *et al.*, "The effects of multiview depth video compression on multiview rendering," *Signal Process., Image Commun.*, vol. 24, nos. 1–2, pp. 73–88, Jan. 2009.

[15] Y. Zhang, X. Yang, X. Liu, Y. Zhang, G. Jiang, and S. Kwong, "High-efficiency 3D depth coding based on perceptual quality of synthesized video," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5877–5891, Dec. 2016.

[16] M. S. Farid, M. Lucenteforte, and M. Grangetto, "Evaluating virtual image quality using the side-views information fusion and depth maps," *Inf. Fusion*, vol. 43, pp. 47–56, Sep. 2018.

[17] Y. Zhao, C. Zhu, Z. Chen, D. Tian, and L. Yu, "Boundary artifact reduction in view synthesis of 3D video: From perspective of texture-depth alignment," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 510–522, Jun. 2011.

[18] P. Merkle, K. Müller, D. Marpe, and T. Wiegand, "Depth intra coding for 3D video based on geometric primitives," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 3, pp. 570–582, Mar. 2016.

[19] J. Lee, M. Park, and C. Kim, *3D-CE1: Depth IntraSkip (DIS) Mode*, document JCT3V-K0033, Switzerland, Feb. 2015.

[20] H. Liu and Y. Chen, "Generic segment-wise DC for 3D-HEVC depth intra coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 3219–3222.

[21] H.-B. Zhang, C.-H. Fu, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Efficient depth intra mode decision by reference pixels classification in 3D-HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 961–965.

[22] H.-B. Zhang, S.-H. Tsang, Y.-L. Chan, C.-H. Fu, and W.-M. Su, "Early determination of intra mode and segment-wise DC coding for depth map based on hierarchical coding structure in 3D-HEVC," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Dec. 2015, pp. 374–378.

[23] K.-K. Peng, J.-C. Chiang, and W.-N. Lie, "Low complexity depth intra coding combining fast intra mode and fast CU size decision in 3D-HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1126–1130.

[24] Z. Gu, J. Zheng, N. Ling, and P. Zhang, *Fast Intra SDC Coding for 3D-HEVC Intra Coding*, document JCT3V-I0164, ISO/IEC TC 1/SC 29/WG 11, Japan, Jul. 2014.

[25] Q. Zhang, Q. Yang, Y. Chang, W. Zhang, and Y. Gan, "Fast intra mode decision for depth coding in 3D-HEVC," *Multidimensional Syst. Signal Process.*, vol. 28, no. 4, pp. 1203–1226, Oct. 2017.

[26] Z. Gu, J. Zheng, N. Ling, and P. Zhang, *Fast Intra Prediction Mode Selection for Intra Depth Map Coding*, document JCT3V-E0238, ISO/IEC JTC1/SC29/WG11, Vienna, Austria, Aug. 2013.

[27] G. Sanchez, M. Saldanha, G. Balota, B. Zatt, M. Porto, and L. Agostini, "Complexity reduction for 3D-HEVC depth maps intra-frame prediction using simplified edge detector algorithm," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 3209–3213.

[28] H.-B. Zhang, Y.-L. Chan, C.-H. Fu, S.-H. Tsang, and W.-C. Siu, "Quadtree decision for depth intra coding in 3D-HEVC by good feature," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 1481–1485.

[29] G. Sanchez, B. Zatt, M. Porto, and L. Agostini, "A real-time 5-views HD 1080p architecture for 3D-HEVC depth modeling mode 4," in *Proc. 27th Symp. Integr. Circuits Syst. Design (SBCCI)* Sep. 2014, pp. 1–6.

[30] G. Sanchez, C. Marcon, and L. Agostini, "Real-time scalable hardware architecture for 3D-HEVC bipartition modes," *J. Real-Time Image Process.*, vol. 13, no. 1, pp. 71–83, 2017.

[31] G. Sanchez, L. Agostini, and C. Marcon, "Energy-aware light-weight DMM-1 patterns decoders with efficiently storage in 3D-HEVC," in *Proc. 29th Symp. Integr. Circuits Syst. Design (SBCCI)*, Aug./Sep. 2016, pp. 1–6.

[32] *3D-HEVC Test Model*. Accessed: Sep. 2017. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-16.0/

[33] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[34] C. E. Rhee, K. Lee, T. S. Kim, and H.-J. Lee, "A survey of fast mode decision algorithms for inter-prediction and their applications to High Efficiency Video Coding," *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1375–1383, Nov. 2012.

[35] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.

[36] K. Müller *et al.*, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3366–3378, Sep. 2013.

[37] G. Balota, M. Saldanha, G. Sanchez, B. Zatt, M. Porto, and L. Agostini, "Overview and quality analysis in 3D-HEVC emergent video coding standard," in *Proc. IEEE 5th Latin Amer. Symp. Circuits Syst. (LASCAS)*, Feb. 2014, pp. 1–4.

[38] G. J. Sullivan, J. M. Boyce, Y. Chen, J.-R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of High Efficiency Video Coding (HEVC)," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

[39] W. Ahmad, M. Martina, and G. Masera, "ComplexComplexity and implementation analysis of synthesized view distortion estimation architecture in 3D High Efficiency Video Coding," in *Proc. Int. Conf. 3D Imag. (IC3D)*, Dec. 2015, pp. 1–8.

[40] G. Sanchez, R. Cataldo, R. Fernandes, L. Agostini, and C. Marcon, "3D-HEVC depth maps intra prediction complexity analysis," in *Proc. IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Dec. 2016, pp. 1–4.

[41] G. K. Tam *et al.*, "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 7, pp. 1199–1217, Jul. 2013.

[42] P. A. A. Assuncao, "Evolution from 3D video to light-field coding and transmission over future media networks," in *Proc. 18th Medit. Electrotech. Conf. (MELECON)*, Apr. 2016, pp. 1–5.

[43] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG pleno: Toward an efficient representation of visual reality," *IEEE Multimedia*, vol. 23, no. 4, pp. 14–20, Oct./Dec. 2016.

[44] D. Marpe *et al.*, "Video compression using nested quadtree structures, leaf merging, and improved techniques for motion representation and entropy coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1676–1687, Dec. 2010.

[45] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1697–1706, Dec. 2012.

[46] M. Wien, "High efficiency video coding: Coding tools and specification," in *Signals and Communication Technology*. New York, NY, USA: Springer, 2015, p. 339.

[47] M. T. Pourazad, C. Doutre, M. Azimi, and P. Nasiopoulos, "HEVC: The new gold standard for video compression," *IEEE Consum. Electron. Mag.*, vol. 1, no. 3, pp. 36–46, Jul. 2012.

[48] H. Schwarz *et al.*, "3D video coding using advanced prediction, depth modeling, and encoder control methods," in *Proc. Picture Coding Symp. (PCS)*, May 2012, pp. 1–4.

[49] L. Zhang *et al.*, "Inter-view motion prediction in 3D-HEVC," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 17–20

[50] Y. Yan, H. Li, and M. M. Hannuksela, "Multiview-video-plus-depth coding and inter-component prediction in high-level-syntax extension of H.265/HEVC," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2013, pp. 406–409.

[51] S. Sugimoto, S. Shimizu, and A. Kojima, "Improved disparity vector derivation for inter-view residual prediction in 3D-HEVC," in *Proc. IEEE Vis. Commun. Image Process. Conf. (VCIP)*, Dec. 2014, pp. 77–80.

[52] L. Zhao, L. Zhang, S. Ma, and D. Zhao, "Fast mode decision algorithm for intra prediction in HEVC," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2011, pp. 1–4.

[53] G. Sanchez, L. Jordani, C. Marcon, and L. Agostini; , "DFPS: A fast pattern selector for depth modeling mode 1 in three-dimensional high-efficiency video coding standard," *J. Electron. Imag.*, vol. 25, no. 6, p. 063011, Nov. 2016.

[54] G. Tech, H. Schwarz, K. Müller, and T. Wiegand, "3D video coding using the synthesized view distortion change," in *Proc. Picture Coding Symp. (PCS)*, May 2012, pp. 25–28.

[55] G. Tech, K. Wegner, Y. Chen, and S. Yea, *3D-HEVC Test Model 1*, document A1005, ITU-T/ISO/IEC JCT3V, , Jul. 2012.

[56] M. Winken, P. Helle, D. Marpe, H. Schwarz, and T. Wiegand, "Transform coding in the HEVC test model," in *Proc. 18th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2011, pp. 3693–3696.

[57] M. Budagavi, A. Fuldseth, and G. Bjøntegaard, "HEVC transform and quantization," in *High Efficiency Video Coding (HEVC): Algorithms Architectures*. Cambridge, MA, USA: Springer, 2014, pp. 141–169.

[58] X. Zhao, L. Zhang, Y. Chen, and M. Karczewicz, "CE6.Hrelated: Simplified DC predictor for depth intra," document JCT3V-D0183, Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V), 4th JCT-3V Meeting, Incheon, South Korea, Apr. 2013.

[59] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[60] D. Rusanovskyy, K. Müller, and A. Vetro, *Common Test Conditions of 3DV Coreexperiments*, document MPEG2011/N12745, ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, Jan. 2013.

[61] Y.-S. Ho, E.-K. Lee, and C. Lee, *Multiview Video Test Sequence and Camera Parameters*, document MPEG2008/M15419, ISO/IECJTC1/SC29/WG11, Archamps, France, Apr. 2008.

[62] J. Zhang, R. Li, H. Li, D. Rusanovskyy, and M. Hannuksela, "Ghost Town Fly 3DV sequence for purposes of 3DV standardization," document M20027, ISO/IEC JTC1/SC29/WG11 MPEG, Geneva, Switzerland, Mar. 2011.

[63] M. Domañski *et al.*, *Poznań Multiview Video Test Sequences and Camera Parameters*, document MPEG 2009/M17050, ISO/IEC JTC1/SC29/WG11, Xian, China, Oct. 2009.

[64] D. Rusanovskyy, P. Aflaki, and M. Hannuksela, *Undo Dancer 3DV Sequence for Purposes of 3DV Standardization*, document M20028, ISO/IEC JTC1/SC29/WG11 MPEG, Geneva, Switzerland, Mar. 2011.

[65] National Institute of Information and Communication Technology. *NICT-3D CG Test Sequences (Shark and MicroWorld)*. Accessed: Sep. 2017. [Online]. Available: http://ftp.merl.com/pub/tian/NICT-3D/Shark

Gustavo Sanchez received the B.S. degree in computer science from Federal University of Pelotas in 2012, the Electrical Engineer degree from Sul-Rio-Grandense Federal Institute of Education, Science and Technology in 2013, and the M.Sc. degree in computer science from Federal University of Pelotas in 2014. He is currently pursuing the Ph.D. degree in computer science with Pontifical Catholic University of Rio Grande do Sul. He has been a Professor with IF Farroupilha, Brazil, since 2014. He has research experience on designing algorithms and hardware architectures for video coding.

Jarbas Silveira (M'16) received the Ph.D. degree in teleinformatics engineering from Federal University of Ceará (UFC) in 2015. He has been an Adjunct Professor with the Teleinformatics Department, UFC, Brazil, since 2009, where he is also with the Engineering Laboratory Computer Systems. His research interests are in the areas of embedded systems on digital circuits, computer architecture, on-chip communication architectures, fault tolerance, and real-time systems.

Luciano V. Agostini (M'06–SM'11) received the M.S. and Ph.D. degrees from Federal University of Rio Grande do Sul, Porto Alegre, Brazil, in 2002 and 2007, respectively. Since 2002, he has been a Professor with Federal University of Pelotas (UFPel), Brazil, where he leads the Group of Architectures and Integrated Circuits. Since 2013, he has been the Executive Vice President for Research and Graduate Studies of UFPel. He has over 200 published papers in journals and conference proceedings. His research interests include 2D and 3D video coding, algorithmic optimization, arithmetic circuits, FPGA-based design, and microelectronics. He is a Member of the IEEE CAS, CS, and SPS societies. He is also a Member of the Multimedia Systems & Applications Technical Committee at the IEEE CAS. He is also a Member of the ACM, SBC, and SBMicro.

César Marcon (M'15) received the Ph.D. degree in computer science from Federal University of Rio Grande do Sul, Brazil, in 2005. He has been a Professor with the School of Computer Science, Pontifical Catholic University of Rio Grande do Sul (PUCRS), Brazil, since 1995. He is advisor of M.Sc. and Ph.D. graduate students at the Graduate Program in computer science of PUCRS. He has authored over 100 papers in prestigious journals and conference proceedings. Since 2005, he has 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. He is member of the Brazilian Computer Society.