Real-Time Simplified Edge Detector Architecture for 3D-HEVC Depth Maps Coding

[1,2]Gustavo Sanchez, [3]Mário Saldanha, [3]Marcelo Porto, [3]Bruno Zatt, [3]Luciano Agostini, [1]César Marcon
[1]Pontifical Catholic University of Rio Grande do Sul – Porto Alegre, Brazil
[2]Instituto Federal Farroupilha– Alegrete, Brazil
[3]Federal University of Pelotas – Pelotas, Brazil
gustavo.sanchez@acad.pucrs.br, {mrdfsaldanha, porto, zatt, agostini}@inf.ufpel.edu.br, cesar.marcon@pucrs.br

***Abstract*—This paper introduces the Simplified Edge Detector (SED) architecture for 3D-HEVC depth maps coding. The SED algorithm classifies the encoding block as homogeneous or edge. When SED classifies the encoding block as homogeneous, the encoding is simplified skipping the bipartition modes evaluation. This approach is capable of reducing 96.7% the bipartition modes evaluation with a drawback of only 0.94% in BD-rate. The SED hardware is capable of providing the decision for all available blocks inside a 32×32 block in only 34 cycles, with a power dissipation of only 25 µW per frame, when synthesized for 65 nm ST standard cells technology.**

Keywords—3D-HEVC; Intra-prediction mode; Edge Detector; Hardware design

I. Introduction

The 3D-High Efficiency Video Coding (3D-HEVC) [1][2] adopts the Multiview Video plus Depth (MVD) [3] format, where depth information is used in addition to texture (regular data seen in video sequences); i.e., a depth map is associated with a corresponding texture frame. Depth maps are captured jointly with texture frames from cameras with infrared sensors. These maps provide geometrical information between objects and the camera, where darker shades of gray (values near of 0) represent far objects, whereas lighter shades of gray (values near of 255) express near objects.

In contrast to texture frame, which presents a complex behavior with sudden variations, a depth map offers a regular behavior with large homogeneous regions and sharp edges, as Fig. 1 depicts. Frequently, objects and the scenario's background contain homogeneous regions, whereas object borders comprise sharp edges.

Fig. 1. Edges (1-4) and homogeneous regions (5-8) of a depth map.

The MVD format is capable of reducing the bandwidth to transmit a 3D video by decreasing the number of texture views to be transmitted. Instead of encoding all cameras present in a scenario, using MVD format it is necessary to encode only a subset of those cameras (texture frames) along with their depth maps. Adaptively, the decoder is capable of generating virtual views located between transmitted views by using Depth Image Based Rendering (DIBR) [4].

The depth maps coding process requires to preserve edges information since errors on the encoded edges can result in the insertion of visual artifacts on the synthesized views. Depth Modeling Modes (DMM) [5], also called bipartition modes, have been designed as an alternative (to HEVC Intra-prediction modes) to encode depth maps in 3D-HEVC, representing sharp edges efficiently.

Real-time encoding of 2D videos is already a challenging process. When dealing with 3D videos, the complexity grows linearly according to the number of cameras used to capture the video. Thus, a considerable higher computational effort is necessary to deal with this amount of data if a large set of views are used, having situations that more than 100 views are used [6]. According to [7], depth maps coding represents about 47.5% of the entire computational effort of the 3D-HEVC. The development of efficient hardware architectures to achieve energy reduction is necessary since 3D video coding is an expensive computational task.

Zhang et al. [8], Gu et al. [9] and Sanchez et al. [10] proposed techniques to reduce the complexity of the 3D-HEVC encoding. Zhang et al. [8] decrease the number of modes evaluated by HEVC intra-prediction and simplifies the DMM-1 process in depth maps coding. Gu et al. [9] introduced a complexity reduction technique that was adopted by 3D-HEVC reference software (more details in Section II). Sanchez et al. [10] proposed an aggressive and lightweight complexity reduction technique using the Simplified Edge Detector (SED) algorithm, detailed in Section III. Among these techniques, only the one presented in [10] is capable of performing bipartition modes evaluation without data dependencies in HEVC intra-prediction mode, which is a desired characteristic in a hardware design. This paper presents the development of the SED hardware design. The main contributions of this paper are summarized as follows:

- **Software evaluation in 3D-HEVC Test Model (3D-HTM) 16.0**: Sanchez et al. [10] have developed and evaluated SED in 3D-HTM 7.0, where many complexity reduction algorithms were not inserted in 3D-HTM as the last version of 3D-HTM provides.
- **Subjective Quality**: Here, we present a subjective quality example of the impact in a depth map and a synthesized

view when the SED algorithm is applied, comparing it with the traditional 3D-HTM encoding flow.

- **Hardware Design**: We propose the first hardware design targeting a fast mode decision for 3D-HEVC with synthesis results for 65 nm ST standard cells technology.

II. Depth Maps Intra-Prediction

Fig. 2 illustrates the intra-prediction mode of 3D-HEVC depth maps performing a dataflow model, where a subset of HEVC intra-prediction modes evaluates an encoding depth block. According to the evaluation of the fast intra-prediction mode, the encoding follows one or two flows.

The first flow uses the Rough Mode Decision (RMD) and the Most Probable Modes (MPM) to select a few HEVC intra-prediction modes among 35 valid modes [11]. The selected modes are added to the Rate-Distortion List (RD-List). This flow does not characterize sharp edges regions well, and distortion in the depth map effectively may result in blurring artifacts at the boundary of objects in the synthesized views. As an alternative, 3D-HEVC adds bipartition modes, which perform the second encoding flow. These modes are essential to represent edges and are capable of providing good quality results at the cost of a high computational complexity [7].

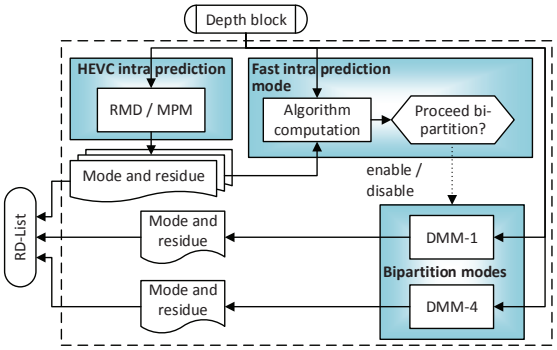


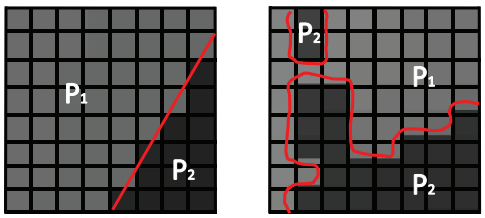
Fig. 2. Intra-prediction mode of 3D-HEVC depth maps performing a dataflow model.

Gu et al. [9] propose a complexity reduction technique that includes bipartition modes in RD-List only if one of the following conditions is true: (i) if planar is the first mode in RD-List, or (ii) if the block variance is higher than an adaptive threshold. If none of these conditions is true, the RD-List is evaluated without the bipartition modes.

After RD-List is complete, the entire list is evaluated by their RD-cost and the mode that obtained the lowest RD-cost is selected as the best encoding mode. The RD-cost evaluation requires performing a complete evaluation of the target mode in the entire encoder. Therefore, by removing a mode from this assessment, it is possible to simplify the encoding process.

The bipartition modes approximate an edge region by a model that divides the area of the block into two regions, represented by constant values, frequently referred to as Constant Partition Value (CPV). The information transmitted for an encoded depth block by bipartition modes is only the partitioning type and the CPV of each region [5].

The bipartition modes implement two partitioning strategies, called wedgelet and contour [5]. In a wedgelet partition, a straight line divides the depth block resulting two regions entirely connected, represented by P1 and P2 in Fig. 3(a). In contrast, for a contour partition, the two regions can be shaped arbitrarily and may consist of multiple disconnected parts, as demonstrated the region P2 in Fig. 3(b).



(a) (b)

Fig. 3. Bipartition modes (a) wedgelet and (b) contour partitioning.

III. Simplified Edge Detector (SED)

A motivational analysis presented in [10] shows that the highest difference of the four corner samples of a block possesses representative information to classify a block into an edge or a constant block. If SED classifies a block as a homogeneous area, only HEVC intra-prediction modes are evaluated, since bipartition modes are designed to encode edges. Otherwise, both HEVC intra-prediction and bipartition modes should be assessed. SED classifies blocks based on a threshold according to the block size and video resolution [10].

In this paper, we present the evaluation of SED algorithm under 3D-HTM 16.0. Table I shows the efficiency of the algorithm with experiments conducted following the Common Test Conditions [12] for all intra mode. The experiments were performed in an AMD OpteronTM Processor 6376 with 64 cores, running at 2.3 GHz, and with 130 GB of RAM.

TABLE I. Efficiency Results – All Intra.

Videos	BD-rate in synthesized views	Time (seconds)		Reduction
		3D-HTM	SED	
Balloons	1.16%	33.37	22.48	32.6%
Kendo	1.10%	38.79	20.22	47.9%
Newspaper	2.46%	51.31	27.78	45.9%
GT_Fly	0.47%	60.88	37.02	39.2%
Poznan_Hall2	1.80%	25.93	19.98	22.9%
Poznan_Street	0.46%	68.35	41.68	39.0%
Undo_Dancer	0.41%	61.03	36.23	40.6%
Shark	0.44%	84.03	54.70	34.9%
Average	**0.94%**	**52.96**	**32.51**	**37.9%**

The SED algorithm increases by 0.94% the BD-rate of synthesized views with the same objective quality. However, SED is capable of providing a time reduction of 37.9% in depth maps, when compared to the standard 3D-HTM 16.0. This result is expressive, mainly when the standard 3D-HTM already applies many complexity reduction techniques.

Fig. 4 displays an evaluation of the bipartition modes skip when using the method proposed in [9] (i.e., in the standard 3D-HTM) and when the SED algorithm is applied. While [9] is capable of providing bipartition modes skips of 50.8%, in average, with variations of more than 15% for different video sequences, the SED algorithm obtains a significant reduction of 96.7%, with maximum variation lower than 5%.

Even SED reducing 96.7% the bipartition modes evaluation, according to our assessments presented in Fig. 4,

when SED runs into a general-purpose architecture, it requires more than 32 seconds per encoding frame (see Table I), in average. Therefore, dedicated hardware designs are necessary for achieving real-time processing high definition videos.

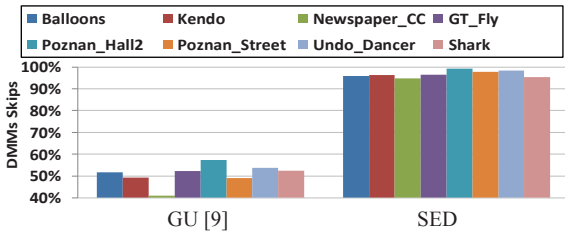
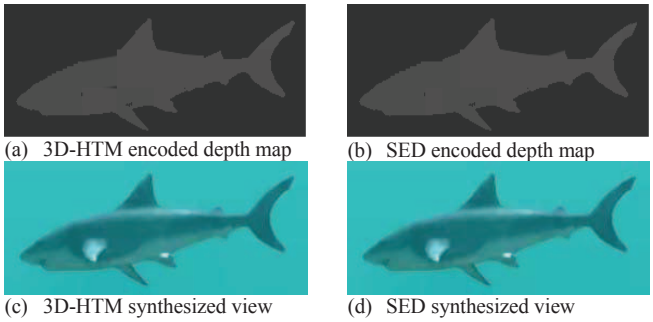


Fig. 4. Comparison of DMMs skips between GU and SED algorithm.

The only works proposing hardware for bipartition modes are [13] (DMM-4) and [14] (DMM-1 and DMM-4). Analyzing the power dissipation presented in [14] (e.g., 166.5 mW for 32×32 blocks), one can conclude that bipartition modes require complexity reduction for real implementations. The SED algorithm eliminates 96.7% of bi-partition modes evaluation, which can help reducing the frequency, and power dissipation of bipartition modes architectures. We also provide a subjective quality example of SED algorithm compared to 3D-HTM. Fig. 5(a) demonstrates the quality of the encoded depth map when encoding Shark 3D video sequence with 3D-HTM while Fig. 5 (b) shows the quality of the same encoded depth map with SED algorithm. As depth maps are not presented to end-users, a synthesized view with both 3D-HTM and SED algorithm (view 3) are shown in Fig. 5(c) and (d), respectively.

There are small differences when comparing Fig. 5(a) and (b). Applying SED algorithm allows smoothing object bodies compared to the standard 3D-HTM (see the middle of the shark in Fig. 5(a) and Fig. 5(b)). When analyzing the synthesized views in Fig. 5(c) and (d), no visual difference can be identified. One can conclude that SED algorithm is capable of reducing 96.7% of undesired DMMs evaluation, smoothing depth maps objects bodies, resulting in a minor impact of synthesized views than traditional 3D-HTM.



(a) 3D-HTM encoded depth map (b) SED encoded depth map

(c) 3D-HTM synthesized view (d) SED synthesized view

Fig. 5. Subjective quality example of SED encoding quality.

IV. Hardware Design

We designed the proposed system for reading a 32 bytes row per cycle, which represents a row of a 32×32 block. In this row, only 16 samples (8-bits wide) should be evaluated to perform the SED decision. Moreover, only 16 rows are required for the SED decision. Consequently, only 256 samples are needed for a 32×32 block to perform the decision to all available blocks. It happens because when evaluating a 32×32 block, there are 84 blocks inside it, being four 16×16, sixteen 8×8 and sixty-four 4×4, when 16, 64 and 256 samples are required for the SED decision, respectively.

Many samples are discarded when the proposed system reads the entire block from the memory. However, the whole block read is essential for allowing the system to be integrated with other encoder modules with shared memory, resulting in a hardware design with no impact in other modules performance. When a specific row contains the information required by the SED algorithm (i.e., one of the four corners of a block), then the useful data is stored in a register. When all information required by a block is already stored, then this block is processed immediately. All used information that is not required by others blocks is discarded, resulting in a reduction of input registers requirements.

Fig. 6 exhibits the proposed system block diagram. The system instantiates 15 classification module architectures, being eight for processing 4×4 blocks, four for 8×8, two for 16×16 and one for 32×32 blocks. Moreover, only 30 1-byte registers are required to store the input information.

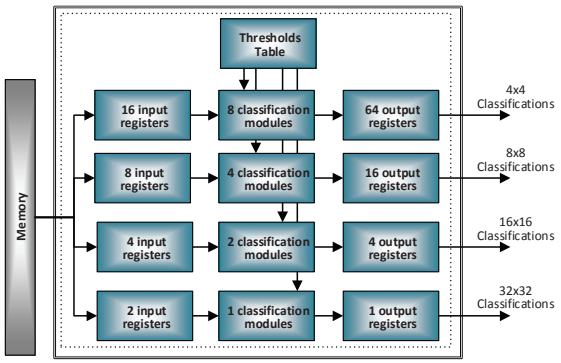


Fig. 6. Complete system design.

Every time the system classifies a block, one 1-bit register (it only stores if the block is constant or an edge) is set with the block classification in the output of the architecture. In a 32×32 block, it is required 64 of these registers to classify the 4×4 blocks, 16 to classify the 8×8 block, four to classify the 16×16 blocks and only one to classify the 32×32 block.

The system requires 34 cycles to process an entire 32×32 block, being 1 cycle for starting the process, 32 cycles to perform the memory read and 1 cycle to its classification.

As presented in [10], SED algorithm requires thresholds according to the encoding block size and frame resolution. In the designed architecture, a module called THRESHOLD TABLE is instantiated, delivering the THRESHOLD information to the classification modules. This design allows easily updating our architecture to a dynamic threshold control version of SED.

Fig. 7 displays the SED classification module architecture, which instantiates six S-COREs. The classification module receives the four border samples, which are distributed among the six available SED Cores (S-CORE), where all combination two by two between border samples are evaluated. If any S-CORE results in the skip decision (i.e., 1), then the OR gates at the rightmost part of the figure also lead to the skip decision.

The S-CORE of the designed architecture is presented in Fig. 8. This module receives two corner samples and the SED

THRESHOLD value corresponding to the encoding block size. The first corner sample (BORDER_1) is subtracted from the second corner sample (BORDER_2), and its result signal is removed in ABS (Absolute) block. Thus, leading to the absolute difference between two evaluated borders.

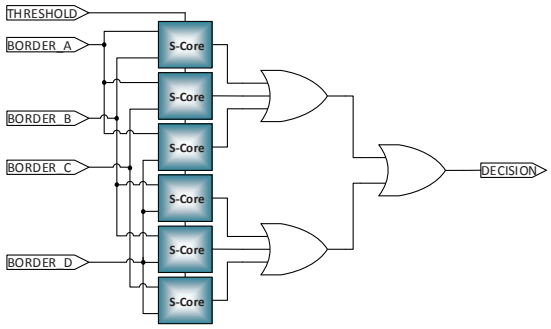


Fig. 7. Classification module block diagram.

A comparator (COMP) detects if this absolute difference is higher than THRESHOLD, when DECISION signal receives 1, meaning that bipartition modes should be evaluated. When the difference is lower than THRESHOLD, then the DECISION signal receives 0, meaning that the encoding block is classified as a homogeneous block, and bipartition mode evaluations are not necessary.

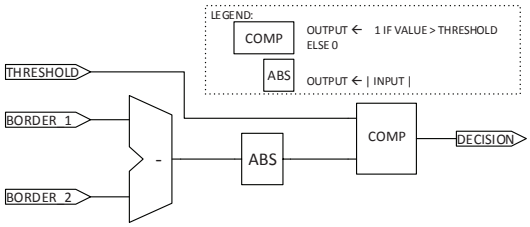


Fig. 8. SED architecture CORE block diagram.

V. Synthesis Results

The SED architecture was described in VHDL and has been synthesized for ST 65nm standard cells technology with the Cadence RTL Compiler tool using 1.0 Volts, 25 ºC and a toggle count of 2%. The synthesis was performed considering six different scenarios; i.e., being capable of processing 1, 5, 10, 25, 50, and 100 HD 1080p views@30fps as there are cases that these quantities of views are required. Synthesis results are presented in Table II.

TABLE II. Synthesis Results for the SED architecture.

Views	Freq. (MHz)	Area (gates)	Power (µW)			
			Leakage	Dynamic	Total	Per view
1	2.1	578	62.5	39	101.5	101.5
5	10.4			176	238.5	47.7
10	20.7			281	343.5	34.4
25	51.7			661	723.5	28.9
50	103.3			1241	1303.5	26.1
100	206.6			2468	2530.5	25.3

For a small set of encoding views, the leakage power (which is constant) represents a significant amount of the total power dissipation. As the number of encoding views increases, the dynamic power starts rising almost constantly, representing a huge amount of total power dissipation, while leakage power almost has no influence. Consequently, the designed SED architecture reduces from 101.5 µW to almost 25 µW per view when a large set of views are encoded.

Analyzing the power dissipation of a bipartition modes architecture found in [14] (e.g., 165.5 mW for 32×32 blocks), it is clear that those 25 µW per view dissipated by the SED architecture is inexpressive when compared to the high power requirement that a bipartition modes architecture without simplification would require.

VI. Conclusions

This paper describes a hardware architecture for the Simplified Edge Detector (SED) algorithm. The SED algorithm is capable of providing 96.7% of bipartition modes evaluation skip with a drawback of 0.94% BD-rate increase. A subjective quality example shows that SED results in the same quality than the standard 3D-HTM solution for synthesized views. The designed architecture is capable of providing the SED decision for all available block sizes inside a 32×32 block in only 34 cycles. Finally, the synthesis results of the SED algorithm for ST 65nm standard cells technology show that we proposed and achieved an efficient low power architecture.

References

[1] G. Sullivan et al., "Standardized Extensions of High Efficiency Video Coding (HEVC)", IEEE Journal of Selected Topics in Signal Processing, vol. 7, n. 6, pp. 1001-1016, Dec. 2013.

[2] G. Tech et al., "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, n. 1, Jan. 2016.

[3] K. Müller et al., "3D High-Efficiency Video Coding for multi-view video and depth data", IEEE IEEE Transactions on Image Processing, vol. 22, n. 9, pp. 3366-3378, Sep. 2013.

[4] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV", Proc. SPIE, Stereosc. Displays Virt. Reality Syst. XI, vol. 5291, pp. 93-104, May 2004.

[5] K. Müller et al., "3D video coding with depth modeling modes and view synthesis optimization", Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, pp. 1-4. 2012.

[6] M. Tanimoto, "Freeviewpoint TV", IEEE Signal Processing Magazine, vol. 28, no. 1, pp.67-76, Jan. 2011.

[7] M. Saldanha et al., "Complexity reduction for 3D-HEVC depth maps coding", IEEE International Symposium on Circuits and Systems (ISCAS), pp. 621-624, 2015.

[8] Q. Zhang et al., "Fast intra mode decision for depth coding in 3D-HEVC", Multidimensional Systems and Signal Processing, pp 1-24, 2016.

[9] Z. Gu et al., "Fast Intra Prediction Mode Selection for Intra Depth Map Coding", ISO/ IEC JTC1/SC29/WG11, Vienna, Aug., 2013.

[10] G. Sanchez et al., "Complexity reduction for 3D-HEVC depth maps intra-frame prediction using simplified edge detector algorithm", IEEE International Conference on Image Processing, pp. 3209-3213, 2014.

[11] J. Lainema, K. Ugur, "Angular intra prediction in high efficiency video coding (HEVC)", in IEEE International Workshop on Multimedia Signal Process (MMSP), pp. 1-5, 2011.

[12] D. Rusanovskyy, K.Muller, A.Vetro, "Common test conditions for 3DV experimentation", ISO/IEC JTC1/SC29/WG11 MPEG2011/N12745, Geneva, Switzerland, Apr. 2012.

[13] G. Sanchez et al., "A real-time 5-views HD 1080p architecture for 3D-HEVC Depth Modeling Mode 4", Symposium on Integrated Circuits and Systems Design (SBCCI), pp. 1-6, 2014.

[14] G. Sanchez et al. "Real-time scalable architecture for 3D-HEVC bipartition modes," Journal of Real-time Image Processing (JRTIP), 2016.