FAST 3D-HEVC DEPTH MAPS INTRA-FRAME PREDICTION USING DATA MINING

[1]*Mário Saldanha,* [2,3]*Gustavo Sanchez,* [2]*César Marcon,* [1]*Luciano Agostini*

[1]Federal University of Pelotas – Pelotas, Brazil
[2]Pontificial Catholic University of Rio Grande do Sul – Porto Alegre, Brazil
[3]IF Farroupilha – Alegrete, Brazil

ABSTRACT

This paper presents a fast 3D-High Efficiency Video Coding (3D-HEVC) depth maps intra-frame prediction based on static Coding Unit (CU) splitting decisions trees. This coding approach uses data mining to extract the correlation among the encoder context attributes and to define a split decision tree for each CU level of the depth maps encoding. The decision trees were trained using the information extracted from 3D-HEVC Test Model (3D-HTM) and using the Common Test Conditions (CTC). Each decision tree defines if the current CU must be split into smaller sizes, considering the encoding context through the evaluation of some current encoder attributes. The proposed solution reaches a complexity reduction of 59.0% for depth maps coding with a negligible impact of 0.18% in the encoding efficiency of synthesized views.

Index Terms— 3D-HEVC, Data mining, Complexity reduction, Depth maps, Intra-frame prediction

1. INTRODUCTION

Focusing on increasing the 3D encoding efficiency, the experts of Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) has developed the 3D-High Efficiency Video Coding (3D-HEVC) [1]. The 3D-HEVC is an extension of the HEVC used on 2D videos.

Multiview Video plus Depth (MVD) [2] is an additional feature provided by 3D-HEVC, where each texture frame is associated with a depth map. The depth maps indicate the distance of each object from the camera and are used to generate virtual views through synthesis view techniques [2] at the decoder side. The complexity of a traditional 3D encoder is already high; however, the insertion of depth maps coding, implies a larger computational complexity [3]. Moreover, the development process inserted many new coding tools in depth maps coding focusing on intra-frame prediction. These tools include Depth Modeling Modes (DMMs) [4], Segment-wise Direct Component Coding (SDC) [5], and Depth Intra Skip (DIS) [6].

The depth maps intra-frame prediction provides a flexible quadtree-based structure, where each frame is divided into *Coding Tree Units* (CTUs), and each CTU can be recursively divided into *Coding Units* (CUs) [7]. The maximum and minimum sizes of a CU are 64×64 and 8×8, respectively. For intra-frame prediction, each CU may be divided into one or four *Prediction Units* (PUs) whose sizes vary from 4×4 to 64×64 [7].

In 3D-HEVC Test Model (3D-HTM) [8] the partitioning structure for each CTU is chosen through Rate-Distortion Optimization (RDO), which assesses many combinations of encoding structures (block partitions and prediction modes) seeking for the best encoding possibility. This process reaches a very high coding efficiency at the cost of a significant increase in the encoder computational complexity when compared with previous standards.

Some works proposed solutions to decrease the encoding computational complexity of depth maps intra-frame prediction, such as [9]-[11], and use the execution time as the metric to evaluate this complexity. Our previous work [9] presents a fast block-level decision based on simplified edge detector for skipping unnecessary DMMs evaluations. Peng et al. [10] propose a block-level and a quadtree-level decision algorithm proposing a threshold based on Rate-Distortion (R-D) cost of the prediction modes. The quadtree-level algorithm computes the CU variance and the maximum variance of the sub-blocks. The split only occurs if the maximum variance of sub-blocks is higher than the CU variance or the CU variance is higher than a threshold. Zhang et al. [11] exploit a QP-based quadtree depth limit to detect if the information of smaller blocks is relevant and the current CU must be split.

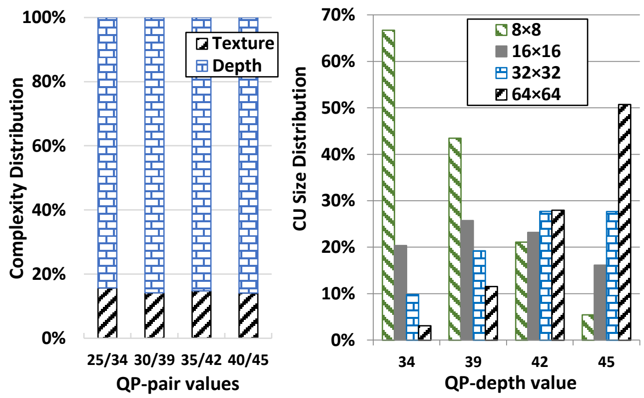
This paper proposes a data mining approach to build static decision trees to define if each CU should be or not split into smaller CUs to predict depth maps intra-frame, without using the full RDO evaluation. Experimental results demonstrate the encoding complexity, regarding processing time, is reduced about 50% maintaining the encoder R-D efficiency. Moreover, the proposed solution surpasses the coding efficiency of related works methods.

2. INITIAL ANALYSIS AND MOTIVATION

This Section presents a first analysis of the 3D-HEVC encoder behavior. These experiments use the 3D-HTM 16.0

version considering the Common Test Conditions (CTC) [12] under All-Intra (AI) encoder configuration [13].

Fig. 1(a) shows the complexity distribution (concerning processing time) between texture and depth maps using the four Quantization Parameter-pairs (QP-pair) values (QP-texture/QP-depth). The worst case of texture coding complexity is about 15.7%, which is much lower than depth map coding complexity. It occurs in the AI configuration because texture coding only applies the HEVC intra-frame prediction, whereas, depth maps coding also uses DMMs, DIS and SDC evaluations [3]. This analysis shows that the depth maps coding is 5.8 times more complex than the texture coding, on average.



(a) (b)

Fig. 1. (a) Complexity distribution for texture and depth coding and (b) CU size distribution for depth coding.

From Fig. 1(a) it is possible to conclude that depth maps encoding complexity is a critical bottleneck in the 3D-HEVC and solutions able to reduce this complexity without penalizing the encoding efficiency are crucial, especially for applications targeting battery-powered devices.

Fig. 1(b) shows the CU size distribution for depth maps coding per QP-depth value, highlighting a variation in the QP-depth causes a different CU size distribution; i.e., the CU size distribution of each encoding block is not homogeneous. It occurs because the QP-depth value defines the compression rate and directly affects the image quality.

The details (and quality) of the image are inversely proportional to the QP-depth value. High QP-depth values generate coded frames with more homogeneous areas (many image details are suppressed), and these areas are efficiently encoded using bigger CUs sizes. Besides, heterogeneous image regions (typically reached with a low QP-depth) must use lower CUs sizes to maintain the encoding efficiency. Based on this fact, for lower QP-depth values, such as 34, about 67% of CUs were encoded with the size of 8×8 and only 3% were encoded with 64×64 CUs. In contrast, with QP-depth=45, about 50% of CUs were encoded with the size of 64×64 and only 5% were encoded with 8×8 CUs. Thus, a solution able to decide when a current CU should or not be split into smaller CUs, considering the QP-depth value, can avoid the excessive cost of the full RDO process.

At runtime, the encoder takes many other decisions that determine the coding efficiency. Consequently, the proposition of solutions that can statically set some of these decisions considering the encoding context (and not the full RDO) and with less impact on the encoder efficiency is highly desirable.

Data mining can be used to correlate the value of dependent variables identifying regularities and building generalizations in attributes of the data set. Decision trees are models built through data mining, which is frequently used when high accuracy and low complexity execution are required [14]. These are essential features for this work since it aims to achieve high complexity reduction maintaining R-D efficiency.

Correa et al. [15] use data mining to evaluate the 2D encoder (texture only) attributes and to extract the knowledge among the correlations of these attributes to avoid some of the dynamic 2D encoder decisions, reducing the encoder complexity. The present work also uses data mining, but to decrease the depth maps encoding complexity in a 3D encoder. Since the scenario is entirely different, a new and complete evaluation of the encoder tools and correlations was necessary, conducting the definition of new attributes (Section 3) and allowing the definition of inedited static CU decision trees (Section 4).

3. ENCODER ATTRIBUTES EVALUATION

Many encoder attributes were evaluated to define the ones most relevant to build the static CU trees for depth maps encoding. The same experimental setup used in Section 2 (considering the CTCs) was employed in this investigation. A significant amount of data from the depth video sequences and internal encoding variables were collected to find features that could lead to sound splitting decisions. The following attributes were evaluated and stored for each CU size during the 3D-HTM encoder execution:

- The current *QP-depth* value, which defines the compression rate and has much impact in the CU split decision.
- *R-D cost* when encoding the current CU size, which was used to better evaluate the relations among the different encoder decisions and the encoding efficiency.
- The variance (*VAR*) of the original samples inside the current CU, which indicates the block homogeneity, and then, can indicate if the block should or not be split.
- The maximum variance of smaller blocks inside of the current CU (*VAR_size*), which represents the maximum variance of the samples inside a block. For a 64×64 CU, there are four instances of this attribute, one for each possible block partition (4×4, 8×8, 16×16 and 32×32). This information can be useful to indicate the homogeneity or the presence of edges into smaller blocks.
- *Average* value of the current CU, which is the average of all samples inside each CU. This information indicates if the encoding CU is near or far from the camera.

Additional details of near objects should be maintained and, in these cases, it is interesting to evaluate lower CUs sizes.

- The single maximum difference between samples of the current CU (*MaxDiff*), which is useful in the CU split decision since this information can indicate sudden variations in samples values.
- The gradient of the four corners of original samples, which is the maximum absolute difference of the four corners in the current CU. This information can be useful to indicate the presence of edges in the current CU (which must be preserved in depth maps [2]).
- Maximum gradient of smaller blocks inside of the current CU, which is the maximum absolute difference of the four corners of these blocks. As *VAR_size*, there are four instances of this attribute for 64×64 CU, and they can indicate when a CU should be split or not.

These attributes were selected to verify depth maps edge regions that tend to be harder to encode [16] and, consequently, tend to cause a splitting decision.

Fig. 2 presents the density probability of the 64×64 CUs do not be split into smaller CUs for some collected attributes. Fig. 2(a) and Fig. 2(b) show that the *MaxDiff* and *VAR_64* have lower values for those CUs that are not split into smaller CUs. The distribution of *R-D cost* is shown in Fig. 2(c) and reveals a correlation with the splitting decision. Fig. 2(d) shows the distribution of *VAR_16*, which provides essential information for sub-blocks inside a 64×64 CU since high values of *VAR_16* can indicate the presence of edges in the current CU and its information can be hidden in generated information for larger blocks. In the case of low variance values in sub-blocks, the current encoding CU tends not to be split into smaller CUs.

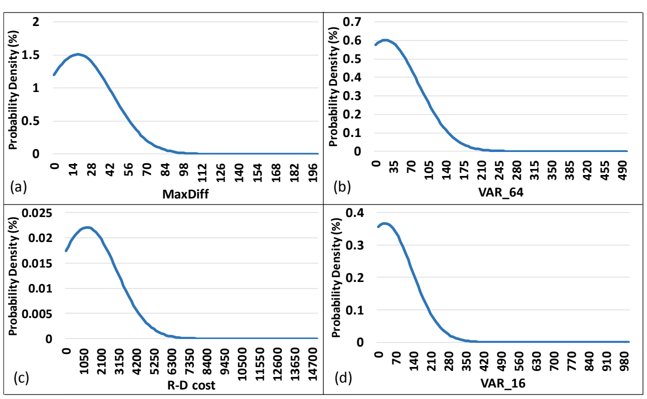


Fig. 2. Probability density of the analyzed attributes does not divide the current 64×64 CUs.

The attributes evaluation allows concluding that only *QP-depth*, *R-D cost*, *VAR, VAR_size*, *Average*, and *MaxDiff* are relevant to build the static CU decision trees. Then, only these attributes were used in the data mining training process, as discussed next in Section 4.

4. PROPOSED CU TREES

Since 3D-HEVC depth maps intra-prediction allows square CU sizes from 8×8 up to 64×64, this work proposes three static decision trees to define when CUs of sizes 16×16, 32×32 and 64×64 should be or not split into smaller CUs.

For the data mining process, Kendo video sequence was encoded in all-intra configuration, considering all CTC QP values. The CTU size has been limited to 16×16, 32×32 and 64×64 pixels for each evaluation. For each encoded CU, were stored: (i) all information that was presented in Section 3 and (ii) the information indicating if the CU has been split or not. It is important to emphasize that there are a limited number of 3D video sequences with their depth maps available to make 3D video coding experiments. Therefore, Kendo video sequence was randomly selected from the CTC dataset, and it was used to extract the data necessary to the offline training process. Only one video sequence was used in the training process to avoid overtraining. However, all video sequences defined in the CTCs were evaluated in Section 5 to demonstrate that the trained solution is capable of achieving a high quality in different encoding scenarios.

The Waikato Environment for Knowledge Analysis (WEKA) [17], version 3.8, was used to train each decision tree. The training was performed using the J48 algorithm, which is an open-source implementation of the C4.5 algorithm [18] available on WEKA. Seeking for a better solution for data balancing, the input files were organized in two sets of data with equal sizes containing inputs that result in (i) splitting and (ii) not splitting of CUs. Besides, to avoid the overfitting problem on the train data set, the *Reduced Error Pruning* (REP) [19] was performed in each tree, reducing the depth of decision trees and allowing a better generalization.

Fig. 3 illustrates the static decision tree generated for 64×64 CUs, where the leaves "N" and "S" correspond to the not split and split decisions, respectively. The decision trees for 32×32 and 16×16 CUs are composed of five and eight decision levels, respectively. The attributes in each decision tree were selected through the information gain, which is used by WEKA decision trees training algorithm [18].

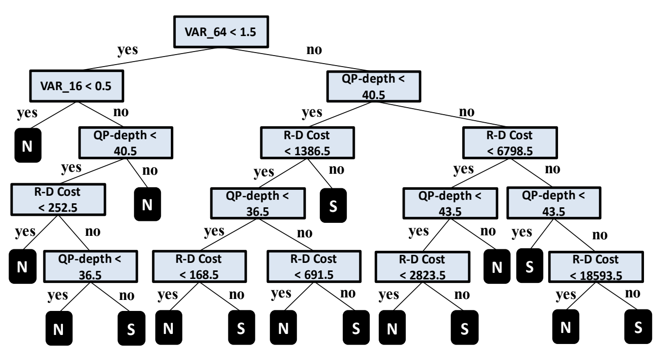


Fig. 3. Decision tree for splitting decision in 64×64 CUs.

Table I presents the complete list of attributes and the corresponding usage in the three proposed decision trees (details about the attributes can be seen in Section 3). One can notice that the proposed solution does not add any new computational complexity to the 3D-HEVC encoder since the training is an offline operation, which is performed only once to define the static trees, and the attributes are easily obtained.

Table I. Attributes used in decision trees.

Decision trees	Attributes								
	QP-depth	*R-D cost*	*Var*					*MaxDiff*	*Average*
			4	*8*	*16*	*32*	*64*		
16×16	×	×	×	×	×			×	×
32×32	×	×		×		×			×
64×64	×	×			×		×		

5. EXPERIMENTAL RESULTS

The three proposed static decision trees were implemented into the 3D-HEVC Test Model (3D-HTM) (version 16.0) [8] and evaluated following the CTC for 3D experiments [12] in all-intra encoder configuration, aiming to assess the performance of the proposed solution.

Table II presents the results of this solution regarding Bjontegaard Delta-rate (BD-rate) [20] considering the synthesized views quality, the complexity reduction (regarding processing time) for the whole 3D-HEVC encoder (texture and depth maps) and the percentage of not split decisions reached for each decision tree.

The proposed solution was able to achieve an average complexity reduction of 52.4% (from 37.3% to 63.4%). Considering only the depth maps coding, the proposed solution can achieve an average complexity reduction of 59%. The average BD-rate degradation was of 0.18% (from 0.04% to 0.62%).

The complexity reduction results were reached because the proposed decision trees, on average, did allow the splitting of 35% of 16×16 CUs, 50% of 32×32 CUs and 60% of 64×64 CUs.

According to the presented experimental results, higher QP-depth values achieved the highest not splitting percentages. An average of 85% of the 64×64 CUs was not split for QP-depth=45 since this QP-depth causes higher compression rates and tends to encode the CTUs with larger CU sizes. These results demonstrate that the proposed solution can achieve a high complexity reduction by removing smaller CU evaluations with small impacts on the encoder R-D efficiency.

It is important to notice that only the Kendo video sequence was used in the offline training process and for the decision trees creation. Moreover, the remaining test sequences were evaluated, and the reached results in complexity reduction and BD-rate degradation were considered promising. Therefore, these results demonstrate that the proposed decision trees were not overfitted for the experimental analysis.

Moreover, the proposed solution was compared with the works [10] and [11], whose results are also displayed in Table II. The method proposed by [10] achieves a complexity reduction of 37.6% with an average BD-rate increase of 0.8%. In [11], a complexity reduction of 41% was obtained with an impact in BD-rate of 0.44%. Therefore, the solution proposed in this work was able to reach the highest complexity reduction when compared with related works. The presented solution also reached the lowest encoding efficiency degradation among the related works with a BD-rate 4.4 times lower than [10] and 2.4 times lower than [11].

6. CONCLUSIONS

This paper presented a fast 3D-HEVC depth maps intra-frame prediction based on static CU splitting decisions trees using data mining. Three static decision trees were trained using WEKA software to define if a current encoding CU should or not be split into smaller CUs. An evaluation of the most relevant encoder attributes was done, where some of these attributes were selected to be used in an offline training. The static decision trees were implemented in the 3D-HTM 16.0 and evaluated under the CTCs for 3D experiments using the all-intra configuration. Experimental results demonstrated that the proposed solution reached a complexity reduction of 52.4% considering the texture and depth maps complexity and 59% when considering only depth maps, with an impact of 0.18% on BD-rate increase of the synthesized views, reaching the best results in both axes when compared with related works.

Table II. Proposed solution results for CTC evaluation in all-intra configuration.

Video	This work					Peng [10]		Zhang [11]	
	CUs not splitting			BD-rate	Complexity reduction	BD-rate	Complexity reduction	BD-rate	Complexity reduction
	16×16	32×32	64×64						
Balloons	47.7%	48.1%	51.1%	0.14%	45.7%	1.2%	27.6%	0.29%	41%
Kendo	45.6%	54.3%	55.9%	0.19%	49.8%	0.7%	26.3%	0.29%	39%
Newspaper_CC	28.2%	40.0%	28.1%	0.11%	37.3%	1.1%	26.6%	-0.23%	36%
GT_Fly	30.4%	49.9%	72.7%	0.06%	58.8%	0.3%	45.1%	0.20%	45%
Poznan_Hall2	35.6%	62.1%	85.9%	0.62%	63.4%	1.3%	43.7%	0.33%	48%
Poznan_Street	24.8%	46.0%	57.7%	0.12%	55.8%	1.4%	49.1%	1.16%	40%
Undo_Dancer	39.0%	55.5%	69.9%	0.14%	56.5%	0.6%	49.1%	1.01%	39%
Shark	29.9%	45.9%	61.3%	0.04%	51.9%	0.1%	33.7%	-	-
Average	**35.1%**	**50.2%**	**60.3%**	**0.18%**	**52.4%**	**0.8%**	**37.6%**	**0.44%**	**41%**

7. REFERENCES

[1] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. Rhee, G. Tech, M. Winken, T. Wiegand. "3D High-Efficiency Video Coding for Multi-View Video and Depth Data," *IEEE Transactions on Image Processing (TIP)*, v. 22, n. 9, pp. 3366-3378, Sep. 2013.

[2] P. Kauff, N. Atzpadin, C. Fehn, M. Muller, O. Schreer, A. Smolic, R. Tanger. "Depth Map Creation and Image-based Rendering for Advanced 3DTV Services Providing Interoperability and Scalability," *Signal Processing: Image Communication (SPIC)*, v. 22, n. 2, pp. 217-234, Feb. 2007.

[3] G. Sanchez, L. Agostini, C. Marcon. "3D-HEVC depth maps intra prediction complexity analysis", *in Proceedings of the IEEE International Conference on Electronics, Circuits, & Systems (ICECS)*, pp. 1-4, 2016.

[4] P. Merkle, K. Muller, D. Marpe, T. Wiegand. "Depth Intra Coding for 3D Video based on Geometric Primitives," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, v. 26, n. 3, pp. 570-582, Feb. 2015.

[5] H. Liu, Y. Chen, "Generic segment-wise DC for 3D-HEVC depth intra coding," *in Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 3219-3222, 2014.

[6] J. Y. Lee, M. W. Park, C. Kim. "3D-CE1: Depth Intra Skip (DIS) Mode," document JCT3V-K0033, Geneva, Switzerland, Feb. 2015.

[7] G. Sullivan, J. Ohm, W. Han, T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," Transactions on circuits and systems for video technology (TCSVT), v. 22, n. 12, pp. 1649-1668, Dec. 2012.

[8] Y. Chen, G. Tech, K. Wegner, S. Yea. "Test Model 11 of 3D-HEVC and MV-HEVC," document JCT3VK1003 of JCT-3V, Geneva, CH, Feb. 2015.

[9] G. Sanchez, M. Saldanha, G. Balota, B. Zatt, M. Porto, L. Agostini. "Complexity reduction for 3D-HEVC depth maps intra-frame prediction using simplified edge detector algorithm," *in Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 3209-3213, 2014.

[10] K. Peng, J. Chiang, W. Lie. "Low complexity depth intra coding combining fast intra mode and fast CU size decision in 3D-HEVC," *in Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1126-1130, 2016.

[11] H. Zhang, Y. Chan, C. Fu, S. Tsang, W. Siu. "Quadtree decision for depth intra coding in 3D-HEVC by good feature," *in Proceedings of the Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1481-1485, 2016.

[12] D. Rusanovskyy, K. Muller, A. Vetro. "Common Test Conditions of 3DV Core Experiments," ISO/IEC JTC1/SC29/WG11 MPEG2011/N12745, Geneva, Jan. 2013.

[13] F. Bossen. "Common Test Conditions and Software Reference Configurations," JCT-VC document L1100, Jan. 2013.

[14] C. Apté, S. Weiss. "Data mining with decision trees and decision rules," *Future generation computer systems*, v. 13, n. 2, pp. 197-210, 1997.

[15] G. Correa, P. Assuncao, L. Agostini, L. Cruz. "Fast HEVC Encoding Decisions Using Data Mining", *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, v. 25, n. 4, pp. 660-673, 2015.

[16] M. Saldanha, G. Sanchez, B. Zatt, M. Porto, L. Agostini. "Complexity Reduction for 3D-HEVC Depth Maps Coding," *in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 621-624, 2015.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten. "The WEKA data mining software: an update," *SIGKDD Explorations*, v. 11, n. 1, pp. 10-18, 2009.

[18] J. Quinlan. "C4.5: Programs for Machine Learning," *Morgan Kaufmann Publishers*, 1993.

[19] C. Brunk, M. Pazzani. "An investigation of noise-tolerant relational concept learning algorithms," *in Proceedings of the International Workshop on Machine Learning*. pp. 389-393, 1991.

[20] G. Bjontegaard. "Calculation of average PSNR differences between RD-curves," Technical Report VCEG-M33, ITU-T SG16/Q6, Austin, Texas, USA, 2001.