Signal, Image and Video Processing (2019) 13:1079–1086
https://doi.org/10.1007/s11760-019-01450-3

ORIGINAL PAPER

Analysis of parallel encoding using tiles in 3D High Efficiency Video Coding

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

Received: 17 September 2018 / Revised: 12 December 2018 / Accepted: 22 February 2019 / Published online: 4 March 2019

Abstract
This paper analyzes the usage of tiles in 3D-High Efficiency Video Coding. We investigate the encoding impact of some tiles divisions using a Bjontegaard delta rate (BD-rate) on the common test conditions for 3D videos. This analysis demonstrated dividing the video into square tiles increases the encoding performance compared to non-square tiles (about 2.2%). In another analysis, we concluded the workload is highly unbalanced in the uniform distribution contributing to limit the speedup possibilities for the homogeneous multicore systems. Authors are not aware of any work focusing on balancing depth maps workload in tiles; therefore, we pointed out relevant studies and solutions for future projects in this open research area.

Keywords 3D-HEVC · Parallelism · Tiles · Real time

1 Introduction

The Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) [1] developed the 3D High Efficiency Video Coding (3D-HEVC) standard [2] to fulfill the rising demand for 3D video applications. 3D-HEVC enhances the encoding efficiency compared to its predecessor H.264/MVC [3] with a reduced encoding/transmission bit rate for a similar 3D video quality [4]. To achieve such a goal, 3D-HEVC inherits and improves several coding tools from the High Efficiency Video Coding (HEVC) [5], inserts new encoding tools to explore 3D videos and adopts the multiview video plus depth (MVD) [6] data format.

MVD associates with each texture frame, a depth map indicating the distance among objects of the scenario and cameras; a depth map displays the scene from the same point of view of the texture [7]. Although depth maps are not visualized, they are essential for generating the synthesized views at the decoder side. Techniques, such as depth image-based rendering (DIBR) [8], are used for interpolating the texture views based on the depth map information and generating a set of high-quality synthesized views [9]. As a drawback, the depth maps are composed of sharp edges requiring high computational effort for effective encoding.

Figure 1a, b shows the texture frame and its associated depth map, respectively, both extracted from the first frame of the Balloons video [10]. Usually, transitions among texture pixels are smooth; however, a depth map has (1) homogeneous regions, with negligible variations between pixels and (2) sharp edges, placed between homogeneous regions [11] [12]. Since keeping accuracy in the depth map coding is essential for synthesizing high-quality views [13] and considering these different features, new coding tools were required for getting high efficiency in the depth map coding. Among the tools added in the depth map coding, we highlight bipartition modes [14], intra-picture skip (also known as depth intra skip—DIS) [15] and DC-only coding (also known as segment-wise DC coding—SDC) [16].

High-definition videos comprise a vast amount of information hampering their real-time implementation when considering sequential approach and single-core systems. This problem is even worse when encoding a 3D video, requiring to process several views in parallel, rising the encoding effort due to the extra tools used for depth map coding [17]. Some works proposed depth map time-saving techniques for single-core systems [18–23]; however, their speedup gains do not allow real-time 3D video encoding. Our early works explored depth modeling mode one (DMM-1) parallelism under multicore [24] and GPU [25] architectures. Although the proposed implementations achieved real-time DMM-1

✉ Gustavo Sanchez
gustavo.sanchez@acad.pucrs.br

1 PUCRS, Porto Alegre, Brazil

2 IF Farroupilha, Alegrete, Brazil

3 UFPel, Pelotas, Brazil

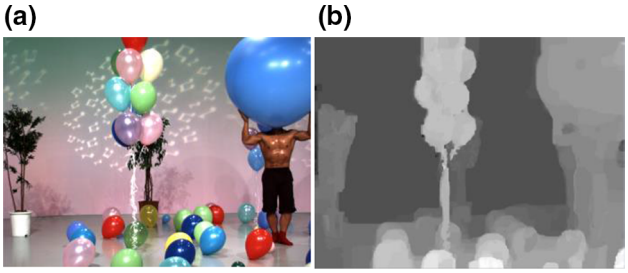

Fig. 1 Balloons video sequence frame with **a** texture view and **b** its associated depth map [10]

coding, the remaining 3D-HEVC tools still need an efficient implementation to reach real-time encoding.

The modern video coding standards have been designed taking specific advantage of the most recent industry technologies focusing on parallel processing systems, such as multicore and distributed systems. During the standardization, the experts introduced three parallelization techniques for HEVC encoding/decoding [26] inherited by 3D-HEVC: (1) slices, (2) wavefront parallel processing (WPP) and (3) tiles.

Misra et al. [27] state the use of tiles is the most promising parallel approach for HEVC. There are some works proposing tile explorations aiming to speed up the encoding [28–30] or enhance the coding efficiency [31]. However, an extensive search of the literature was performed, and no work was found exploring the usage of tiles for 3D systems. Therefore, we propose the first tile analysis for 3D-HEVC, serving as a base for outcoming works in 3D-HEVC real-time encoding design.

Our novel contributions This work is the pioneer considering the use of tiles on 3D videos. We analyze the encoding efficiency and speedup results for 3D-HEVC using tile partitioning with all-intra and random access encoder configurations. This analysis allows identifying behaviors for texture and depth map coding according to some tiling scenarios. Moreover, we are not aware of any work focusing on tiles usage for 3D-HEVC, and we aim to contribute to future research highlighting some key possibilities for efficient parallel video coding. These possibilities take into account the tiling technology has a high potential to allow creating new applications for 3D video coding exploring the advances in the industry with multicore processors.

The remainder of this paper is divided as follows. Section 2 explains the basic concepts of parallel encoding of HEVC-based encoders. Section 3 details the setup and configurations for the experiments performed in this paper. Sections 3.2 and 3.3 present the analysis of the results regarding all-intra and random access configurations, respectively. Based on these analyses, Sect. 4 presents some insights into open research areas using tiles for 3D-HEVC parallel encoding. Finally, Sect. 5 concludes this work.

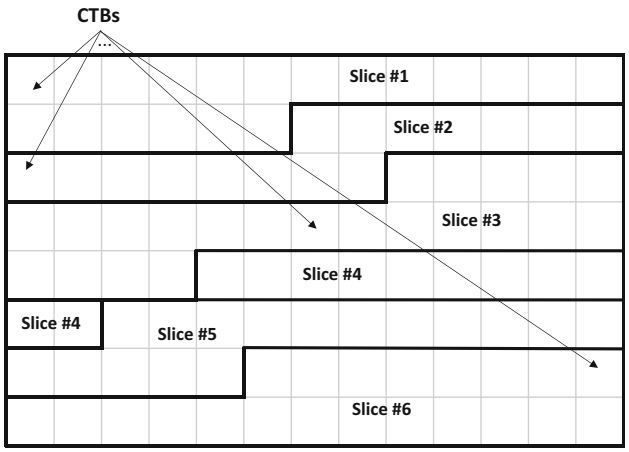

Fig. 2 Example of a slice-based picture partitioning

2 Parallel encoding background

Some features of the 3D videos, such as encoding several views at the same time, are inherently applicable in parallel encodings. However, the fully exploit of the interview redundancy inside a 3D video requires encoding those views sequentially for allowing data reuse, enhancing the encoding efficiency but reducing the potential for parallelism. Thus, 3D-HEVC provides slices, WPP and tiles, which are three parallel techniques.

Slices is a parallel encoding strategy introduced by H.264/AVC [32] to prevent coding losses in the case of transmission errors because a single error may lead to the inability to process a significant portion of the bitstream due to the data dependencies. Slices can also allow a parallelization approach of the encoded/decoded frames, where each slice can be encoded/decoded independently (without information from any other slice).

3D-HEVC partitioning a picture frame into coding tree blocks (CTBs) composing a slice, and CTBs within a slice are processed in raster scan order. Figure 2 exemplifies a slice-based picture partitioning of six slices, where each slice groups different sets of CTBs.

Although the slice approach provides a flexible definition of partition shapes, every slice contains a header, which adds a significant overhead into the bitstream. Thus, many slices incur significant coding losses.

The WPP technique parallelizes each CTB row of a picture without broken dependencies among the encoded blocks. Figure 3 exemplifies CTB rows processing in parallel with WPP enabled, where each thread processes each CTB row. While the first row is processed in raster scan order, the CTBs in the next rows are processed only when two CTBs from the previous row were already encoded. On the one hand, a less encoding loss is noticed; on the other hand, the parallelization

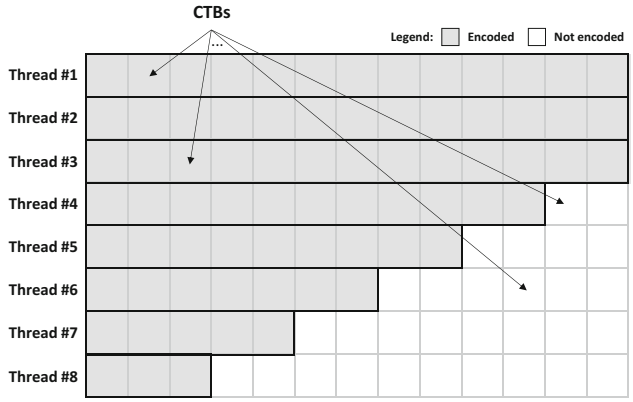

Fig. 3 CTB rows processing in parallel with WPP technique

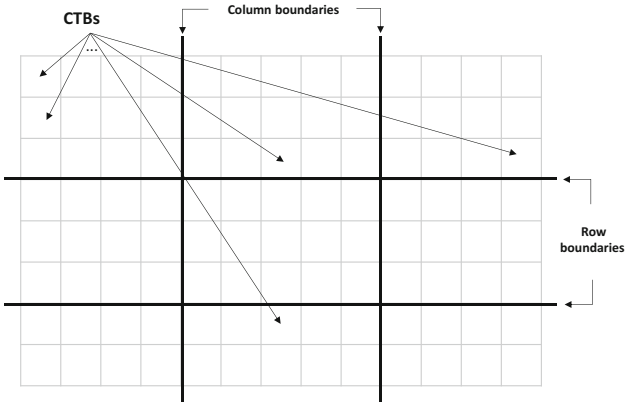

Fig. 4 Illustration of a 3×3 tile partitioning with a non-uniform format

effectiveness is reduced because the parallelism is limited by the encoding of at most one block per row.

The tile approach, like slice boundaries, breaks the inter-block dependencies. Besides, it has the advantage of creating picture partitions that can be processed in parallel without incurring significant coding losses. Among the parallelization possibilities, the tile provides the most flexible structure, being capable of adapting itself according to the available computational resources and allowing exploring several levels of parallelism without significant encoding losses [26].

Each frame can be divided into sets of CTBs limited by rows and columns, and the intersection of a limiting row and column results in a tile. Figure 4 exemplifies a frame partitioning into 3×3 non-uniform tile format, resulting in nine tiles encoded independently. It is important to note the tiles partitioning is flexible and the tiles and CTBs boundaries are always aligned. Thus, the tiles divisions are more compact and group CTBs without breaking a large number of dependencies when compared to the slices partitioning. It occurs because tiles allow partition shapes containing samples with a higher correlation potential than slices. As an additional advantage, tiles divisions do not contain headers for each partition and consequently do not increase the encoded information transmitted to the bitstream. However, similar to slices, encoding efficiency losses are noticed according to the increase in the number of tiles, due to the break of dependencies.

The usage of tile structure allows dividing the encoding frame into a different number of CTBs to compose each tile, providing levels of parallelism. During the encoding, the tile workload is not balanced since different video sequences and frames have different characteristics. Thus, tiles composed of regions with high movement intensity or sudden variation in sample values tend to require a higher encoding computational effort than tiles composed of simpler regions. Thus, uniform distribution results in a non-balanced tiles distribution; consequently, distributing the tiles processing into different cores results in lower speedups since some cores finish their processing later than other cores. Then, we selected tiles approach to be further discussed and evaluated in the remaining of this paper.

Sections 3.2 and 3.3 analyze the encoding efficiency loss using some tile partitions and encoder configurations, and evaluate the speedup possibilities provided by the tile approach.

3 Experimental setup & tiles analysis

3.1 Experimental setup

To evaluate the experimental results, we modified the 3D-HEVC Test Model (3D-HTM) version 16.0 allowing storing some features such as the encoding time of each tile. Our experiments assessed all videos following the common test conditions (CTC) for 3D videos [33]. CTC defines the evaluation of three 1024×768 videos (Balloons, Kendo and Newspaper) and five 1920×1088 videos (Poznan_Hall2, Poznan_Street, Shark, GT_Fly and Undo_Dancer), where each video is coded in four quantization levels (QP-texture/QP-depth) with values (25/34), (30/39), (35/42) and (40/45).

We applied the Bjontegaard delta rate (BD-rate) [34] metric to compare the encoding efficiency loss obtained by each tile partition compared to the encoding without using tiles. Table 1 describes the main configurations of the experiments employed in this paper for both all-intra (Sect. 3.2) and random access (Sect. 3.3) scenarios. These evaluations were executed in a computer with the Ubuntu 12.04 operational system, AMD $Opteron^{TM}$ Processor 6376 with 64 cores running at 1400 MHz and 128 GB DDR3 memory.

Table 1 Configurations used in the experiments

Profile	All-intra	Random access
GOP size	1	8
Intra-period	1	24
Bits per pixel	8	
Texture/depth views	3/3	
QP-pair	(25/34), (30/39), (35/42), (40/45)	
DMM evaluation	Enabled	
Depth intra skip	Enabled	
VSO	Enabled	
Rate control	Disabled	
3D-HTM version	16.0	

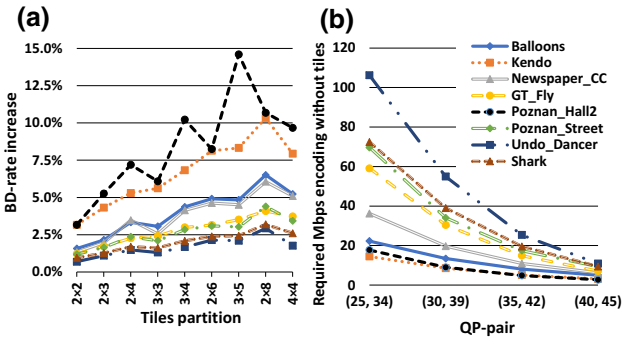

Fig. 5 **a** BD-rate impact for some tile partitions and **b** Mbps required by the encoded video without tiles, considering CTC sequences and QP-pair encoded with all-intra encoder configuration

3.2 Tile analysis in all-intra configuration

3.2.1 BD-rate analysis

Our first analysis compares the BD-rate performance of parallel versions that divide the frames into nine formats of tiles to the version without using tiles. Figure 5a shows these evaluation results, according to the encoded video. Note that although there are several formats of tiles, in each partition the tiles have the same size (i.e., a uniform approach).

Increasing the number of tile partitions raises the parallelization possibilities. However, as Fig. 5a shows, this increase tends to grow the BD-rate. Dividing the frame into square tiles (i.e., 2×2, 3×3 and 4×4) tends to reduce the coding losses per tile partition compared to non-square tiles. It happens because square tiles have a smaller perimeter around their boundaries, consequently breaking fewer dependencies than non-square tiles [27]. This fact can be noted when increasing from 2×4 to 3×3 tiles and from 3×5 to 4×4 tiles, where there is a small BD-rate reduction with a higher possibility of speeding up, due to the increase in the number of tiles partitions. For example, in 2×8 tiles partitioning a significant BD-rate increase is obtained since the tile has a higher perimeter in its boundary than in the 3×5 or 4×4 tiles partitioning.

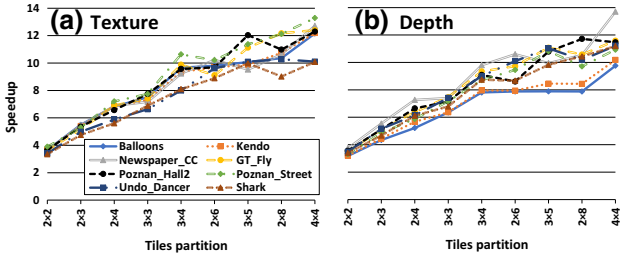

Fig. 6 Speedup results of texture and depth map encoding with uniform tile partitions for all-intra configuration

The better BD-rate performances were obtained in Shark and Undo_Dancer video sequences, while Kendo and Poznan_Hall2 sequences obtained the worst results. The correlations among the pixels of each video explain these results. Kendo and Poznan_Hall2 present high pixel correlation; consequently, the encoded stream that requires a lower quantity of bits in the traditional process tends to have a higher break on dependencies among CTBs when tiles are used. Subsequently, this type of video sequence requires the insertion of a higher quantity of bits due to the tiles overhead and the need to reset the encoder context in the dependencies break. The opposite happens with Shark and Undo_Dancer since these video sequences present low pixel correlation.

Figure 5b shows the required bits encoding applying the traditional 3D-HEVC flow (i.e., without using tiles) when encoding with the same parameters described in Table 1. Note that these results are inversely proportional to the results obtained in Fig. 5a. Thus, demonstrating the videos encoded with a smaller bit rate requirement tends to be more susceptible to higher impacts when using tiles.

3.2.2 Speedup possibilities and workload balancing

Figure 6 shows the average speedup results of dividing the frame uniformly in the same tiles format presented in Sect. 3.2.1. The speedup is computed as the sum of each tile's execution time divided by the time spent in the most time-consuming tile in each frame.

The maximum theoretical gain is $N \times M$, where N and M are the horizontal and vertical number of tiles, e.g., in 2×2, 3×3 and 4×4 tile formats, the maxima theoretical gains are 4, 9 and 16, respectively. These gains would only occur when the tiles' workloads are perfectly balanced to have the same encoding time.

From Fig. 6, one can notice the obtained speedups are still far from the maximum theoretical speedup; however, it is possible to already achieve significant acceleration results. For example, the 2×2 tile format, where the maximum theoretical speedup is 4, produces speedups varying between 3.35

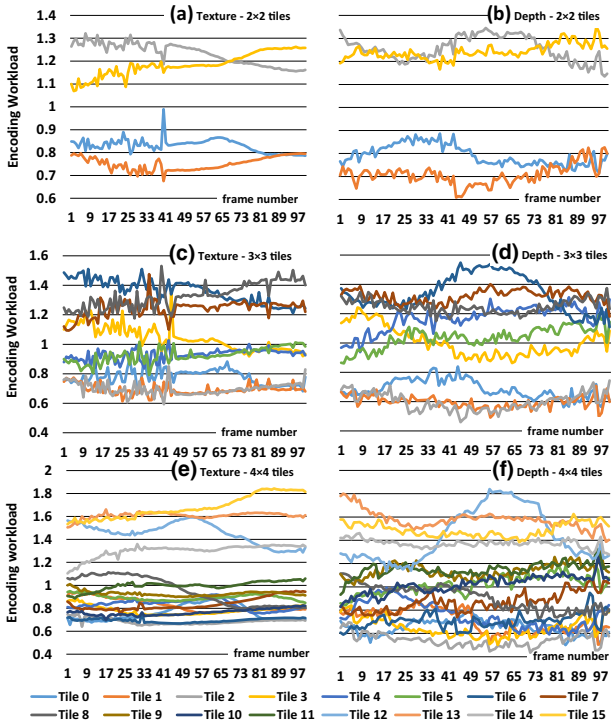

Fig. 7 Encoding workloads of tiles for all-intra encoder configuration: **a** texture—2 × 2, **b** depth map—2 × 2, **c** texture—3 × 3, **d** depth map—3 × 3, **e** texture—4 × 4 and **f** depth map—4 × 4

and 3.90 and between 3.17 and 3.75 for texture and depth map coding, respectively, while the speedups obtained with the 3 × 3 tile format vary from 6.63 to 7.74 for texture coding and from 6.35 to 7.46 for depth map coding, being the maximum theoretical speedup of 9. The speedups attained with the 4 × 4 tile format range from 10.10 to 13.28 and from 9.76 to 13.70 for texture and depth map coding, respectively, again being far from the maximum theoretical speedup of 16.

In most of the cases, the speedup of the texture coding is higher than the speedup of the depth map coding, showing the uniform division of the tiles works better for texture than depth map coding. It happens because the depth map intra-frame prediction has some default context-based speedup decision when encoding its blocks. Their decisions focus on increasing the encoding effort in edge regions and reducing the effort spent in homogeneous regions using the decisions of [35] and [36] that are applied by default in 3D-HTM version 16.0. Thus, there was a higher distortion in the encoding time among the depth maps tiles than texture tiles, contributing to lowering the depth map speedup. Figure 7 shows these speedups with the normalized workload distribution of texture and depth maps for the first 100 frames of the shark video sequence using QP = (25, 34).

The workload was normalized by showing its variation in comparison with the average workload among all tiles. The normalization was performed converting the measured

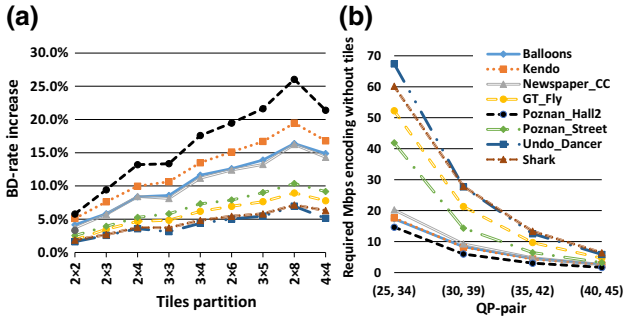

Fig. 8 **a** BD-rate impact for some tile partitions and **b** Mbps required by the encoded video without tiles, considering CTC sequences and QP-pair encoded with random access encoder configuration

execution time on a scale where 1 was the ideal workload. Values higher than 1 indicate that a higher workload was found in that tile partition, while values below 1 indicate that the partition workload could be increased without affecting the speedup.

The workload varies significantly according to the tile being executed since each video region has specific features. Besides, the depth map variation is higher than the texture variation because non-edge regions are encoded efficiently, avoiding the usage of DMMs in the 3D-HEVC reference software by default. Another essential characteristic is that the neighbor frames have similar encoding workload per tile partition; this information can be used as a decision factor for allowing a predictive balancing of tiles in future works.

3.3 Tile analysis in random access configuration

3.3.1 BD-rate analysis

Figure 8a presents a BD-rate analysis of using tiles in random access configuration. A similar behavior discussed for all-intra configuration (Sect. 3.2.1) is noticed, where the square tiles partitions achieve higher encoding efficiency per partition than non-square tiles. As expected, the worse BD-rate results were achieved in Poznan_Hall2 and Kendo video sequences because these sequences are typically encoded with a lower number of bits (Fig. 8b), becoming more susceptible to BD-rate increase when using tiles. Moreover, the same occurs in 2 × 8 tiles partitioning when using RA configuration, where a significant BD-rate increase is obtained since the tile has a higher perimeter in its boundary than in the 3 × 5 or 4 × 4 tiles partitioning.

The random access scenario shows a substantial increase in the BD-rate results, which differs from all-intra configuration analysis. This increase is occasioned mainly due to the break of dependencies in the frames encoded with motion predictions tools, where less information can be explored to produce efficacious predictions. Therefore, the prediction

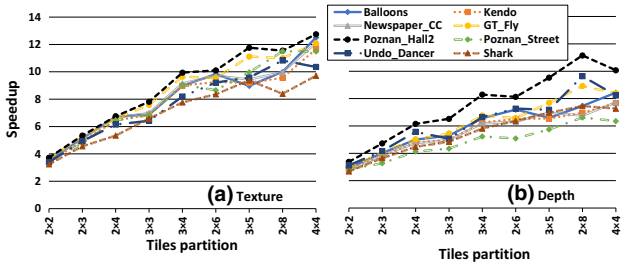

Fig. 9 Speedup results for texture and depth map encoding with uniform tile format and random access scenario

obtained using tiles was worse than the one achieved by using without tiles, contributing to a higher increase in BD-rate results compared with the all-intra scenario.

3.3.2 Speedup possibilities and workload balancing

Figure 9 shows the average speedup of tiling the frame uniformly in some tiles partitioning with random access configuration. Again, the obtained results are still far from the maximum theoretical speedup.

For example, the speedups obtained with the 2×2 tile format vary between 3.24 and 3.72 for texture coding and between 2.68 and 3.38 for depth map coding. The 3×3 tile format produces speedups ranging from 6.42 to 7.81 and from 4.35 to 6.52 for texture and depth map coding, respectively. The speedups attained with the 4×4 tile format range from 9.71 to 12.74 and from 6.36 to 10.07 for texture and depth map coding, respectively. The depth maps reached a slight increase in speedup compared to the texture in the random access configuration. However, for all cases, there is still room for speedup improvement.

Figure 10 details the normalized encoding workload of the first 100 frames of Shark video sequence with QP = (25, 34). This figure shows the texture workload is more stable during the execution when compared to the depth map workload that has abrupt variations between frames. Besides, the premise that neighbor frames have similar encoding workloads, regardless of the tile partition, is also valid for random access configuration. Both the premise and the abrupt variations in the depth map coding between frames are essential characteristics that should be considered when designing a workload balancing system for depth maps.

4 Unexplored research in tiles at 3D-HEVC

From the experiments described in Sects. 3.2 and 3.3, considering the lack in the literature related to the usage of tiles in 3D-HEVC and the fact that the 3D-HEVC coding implies a significantly higher encoding complexity than HEVC, we conclude the parallelism plays an essential role in the design

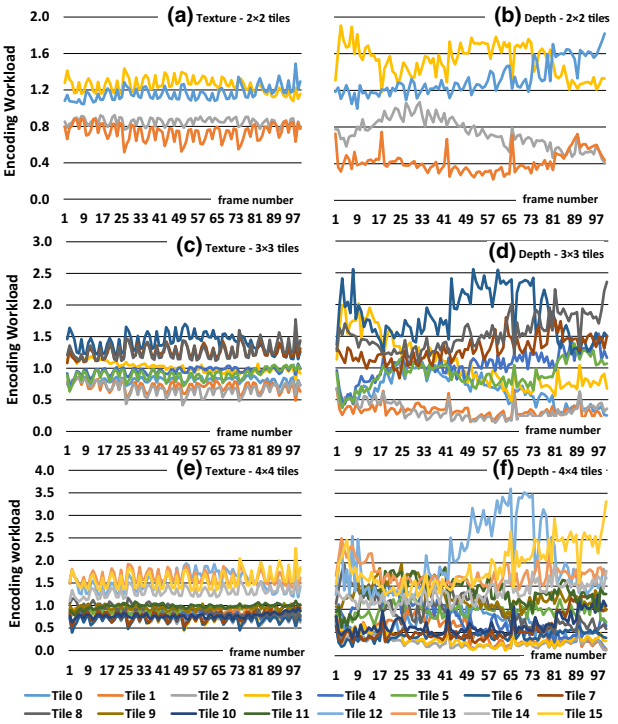

Fig. 10 Encoding workload of different tiles for random access encoder configuration: **a** texture—2×2, **b** depth map—2×2, **c** texture—3×3, **d** depth map—3×3, **e** texture—4×4 and **f** depth map—4×4

of real-time coding systems. Therefore, the research on tiles focusing on 3D-HEVC systems is vital to allow future encoding implementations. We visualize the following possibilities for future researches in tiles for 3D-HEVC:

- *Balancing of the current depth map workload*—our analysis presented in Sect. 3 shows the poor workload balancing obtained in the uniform tile partitions, which is a limiting factor for maximizing the speedup. Therefore, we foresee the possibilities of scientific works in balancing depth map workload by correlating the tiles workload of the associated texture view and other tile workloads of previously encoded depth maps.
- *Exploring the tiles encoding on an asymmetric processor architecture*—the 3D-HEVC encoder can be executed into a system with asymmetric processors, i.e., processors with heterogeneous computing resources. This asymmetric architecture requires techniques of tiles mapping and scheduling for efficient workload balancing. For instance, tiles identified with higher workload can be scheduled to be mapped onto processors with higher performance.
- *Designing tile-based techniques for reducing the memory required for encoding/decoding depth maps*—typically, the entire encoding / decoding frame and reference frames require memory regions to be stored and perform intra- and inter-prediction. However, when encod-

ing/decoding tiles, only some partitions of the frames are required by the memory. So, designing tile-based techniques for reducing the necessary memory of each region can be interesting when designing a dedicated system.
– *Decoding-based workload balancing*—another promising research is exploring the workload balancing in the decoder side because a poor balancing may limit the speedup in the decoding process. It is an open research area (still in 2D video coding) to design predictive tile-based algorithms at the encoder for balancing the tiles workload at the decoder, allowing a higher speedup at the decoder.
– *Controlling the complexity inside each tile by switching encoding algorithms*—a complexity control system can obtain a higher efficiency if it implements a control level that selects among the available encoding algorithms, the one that most efficiently encodes that tile in a given target complexity. This control level is a dynamic approach that can be implemented with a machine learning technique.
– *Exploring regions of interest (ROI) based on the depth map tiles*—tiles are highly used for discovering the ROI inside the video. By using the depth maps, along with the texture, it is possible to design new algorithms for automatically discovering the ROI and increasing the encoding effort specifically in that region, allowing a better subjective video quality.

The use of tile technology has a high potential to meet the demand for computational performance of future applications, especially when newcomer processing systems are being available with an increasing number of cores. Furthermore, since there are no works in the literature proposing solutions using tile technology for depth map coding and given the high complexity associated with the 3D video coding, it is natural to consider that evolution in tile coding systems can allow the creation of new 3D video applications since it can help achieving a high speedup.

5 Conclusions

This paper analyzed the 3D-HEVC parallel encoding using the tile approach with CTC for 3D videos under all-intra and random access configurations. First, we performed a BD-rate analysis among some partition possibilities for each configuration. For both configurations, we obtained the best results for squared tiles partitions; however, in this case, the BD-rate increase is not negligible (ranging from 0.7 to 21.4%), requiring the design of new solutions for raising the encoding efficiency. Besides, the random access scenario presents significant BD-rate increase when compared to the all-intra configuration (7.1% for 4×4 tiles partitioning) since more dependencies among frames applying motion predictions are broken. Another experiment assessed the speedup provided by the tile approach for texture and depth map coding; for both encoder configurations, the speedups were considerably lower than the maximum theoretical speedup.

The encoding workload of the texture frames has been demonstrated more stable than the encoding workload of the depth map frames. It happens because depth map frames have abrupt content variations with well-defined edges and homogeneous regions, while texture frames present a smoother content variation. Finally, we concluded there is significant space for researches applied to the 3D-HEVC tile technology, mainly regarding the depth map coding.

Acknowledgements This paper was performed in cooperation with Hewlett-Packard Brazil Ltda. using incentives of Brazilian Informatics Law (Law n° 8.248 of 1991). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001. Authors also would like to thank CNPq and FAPERGS Brazilian research agencies to support the development of this work.

References

1. JCT-3V. http://phenix.int-evry.fr/jct2/. Online; Access in August (2018)
2. Müller, K., Schwarz, H., Marpe, D., Bartnik, C., Bosse, S., Brust, H., Hinz, T., Lakshman, H., Merkle, P., Rhee, F.H., et al.: 3D high-efficiency video coding for multi-view video and depth data. IEEE Trans. Image Process. **22**(9), 3366–3378 (2013)
3. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Joint Draft 8.0 on Multiview video coding, JVT-AB204 (2008)
4. Balota, G., Saldanha, M., Sanchez, G., Zatt, B., Porto, M., Agostini, L.: Overview and quality analysis in 3D-HEVC emergent video coding standard. In: Proc. IEEE Latin American Symposium on Circuits & Systems (LASCAS), pp. 1–4 (2014)
5. Sullivan, G., Ohm, J.R., Han, W.J., Wiegand, T., et al.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
6. Kauff, P., Atzpadin, N., Fehn, C., Müller, M., Schreer, O., Smolic, A., Tanger, R.: Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability. Signal Process. Image Commun. **22**(2), 217–234 (2007)
7. Bal, C., Nguyen, T.Q.: Multiview video plus depth coding with depth-based prediction mode. IEEE Trans. Circuits Syst. Video Technol. **24**(6), 995–1005 (2014)
8. Fehn, C.: Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. Stereosc. Disp. Virtual Real. Syst. **5291**, 93–104 (2004)
9. Zhang, L., Tam, W.: Stereoscopic image generation based on depth images for 3D TV. IEEE Trans. Broadcast **51**(2), 191–199 (2005)
10. Tanimoto Lab. http://www.tanimoto.nuee.nagoya-u.ac.jp/. Online; Access in September (2018)
11. Merkle, P., Muller, K., Wiegand, T.: Coding of depth signals for 3D video using wedgelet block segmentation with residual adaptation. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2013)
12. Muller, K., Merkle, P., Wiegand, T.: 3-D video representation using depth maps. Proc. IEEE **99**(4), 643–656 (2011)
13. Kang, J., Chung, K.: High-performance depth map coding for 3D-AVC. Signal Image Video Process. **10**(6), 1017–1024 (2016)

14. Merkle, P., Muller, K., Marpe, D., Wiegand, T.: Depth intra coding for 3D video based on geometric primitives. IEEE Trans. Circ. Syst. Video Technol. **26**(3), 570–582 (2015)
15. Lee, J., Park, M., Kim, C.: 3D-CE1: Depth intra skip (DIS) mode, doc. JCT3V-K0033 (2015)
16. Liu, H., Chen, Y.: Generic segment-wise DC for 3D-HEVC depth intra coding. In: IEEE International Conference on Image Processing (ICIP), pp. 3219–3222 (2014)
17. Sanchez, G., Silveira, J., Agostini, L., Marcon, C.: Performance analysis of depth intra coding in 3D-HEVC. IEEE Trans. Circuits Syst. Video Technol. (2018). https://doi.org/10.1109/TCSVT.2018.2865645
18. Saldanha, M., Sanchez, G., Marcon, C., Agostini, L.: Block-level fast coding scheme for depth maps in three-dimensional high efficiency video coding. J. Electron Imaging **27**(1), 010,502 (2018)
19. Zhang, H., Fu, C., Chan, Y., Tsang, S., Siu, W.: Efficient depth intra mode decision by reference pixels classification in 3D-HEVC. In: Proc. IEEE International Conference on Image Processing (ICIP), pp. 961–965 (2015)
20. Sanchez, G., Agostini, L., Marcon, C.: A reduced computational effort mode-level scheme for 3D-HEVC depth maps intra-frame prediction. J. Vis. Commun. Image Represent. **54**, 193–203 (2018)
21. Fu, C., Zhang, H., Su, W., Tsang, S., Chan, Y.: Fast wedgelet pattern decision for DMM in 3D-HEVC. In: Proc. IEEE International Conference on Digital Signal Processing (DSP), pp. 477–481 (2015)
22. Oh, B.: Enhanced zonal search algorithm for motion estimation in depth-map coding. Signal Image Video Process. **12**(3), 523–530 (2018)
23. Jing, R., Zhang, Q., Wang, B., Cui, P., Yan, T., Huang, J.: CART-based fast CU size decision and mode decision algorithm for 3D-HEVC. Signal Image Video Process **13**(2), 209–216 (2019)
24. Sanchez, G., Agostini, L., Sousa, L., Marcon, C.: 3D-HEVC DMM-1 parallelism exploration targeting multicore systems. In: Proc. Symposium on Integrated Circuits and Systems Design (SBCCI), pp. 1–6 (2018)
25. Sanchez, G., Agostini, L., Sousa, L., Marcon, C.: Parallelism exploration for 3D high-efficiency video coding depth modeling mode one. J. Real Time Image Process. (2018). https://doi.org/10.1007/s11554-018-0819-3
26. Chi, C., Alvarez-Mesa, M., Juurlink, B., Clare, G., Henry, F., Pateux, S., Schierl, T.: Parallel scalability and efficiency of HEVC parallelization approaches. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1827–1838 (2012)
27. Misra, K., Segall, A., Horowitz, M., Shilin, X., Fuldseth, A., Minhua, Z.: An overview of tiles in HEVC. IEEE J. Sel. Top Signal Process. **7**(6), 969–977 (2013)
28. Storch, I., Palomino, D., Zatt, B., Agostini, L.: Speedup-aware history-based tiling algorithm for the HEVC standard. In: Proc. IEEE International Conference on Image Processing (ICIP), pp. 824–829 (2016)
29. Koziri, M., Papadopoulos, P., Tziritas, N., Dadaliaris, A., Loukopoulos, T., Khan, S., Xu, C.: Adaptive tile parallelization for fast video encoding in HEVC. In: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1–8 (2016)
30. Shafique, M., Khan, M., Henkel, J.: Power efficient and workload balanced tiling for parallelized high efficiency video coding. In: IEEE International Conference on Image Processing (ICIP), pp. 1253–1257 (2014)
31. Blumenberg, C., Palomino, D., Bampi, S., Zatt, B.: Adaptive content-based Tile partitioning algorithm for the HEVC standard. In: Proc. Picture Coding Symposium (PCS), pp. 185–188 (2013)
32. Wiegand, T., Sullivan, G., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 560–576 (2003)
33. Muller, K., Vetro, A.: Common test conditions of 3DV core experiments, ISO/IEC JTC1/SC29/WG11 MPEG2011/N12745 (2014)
34. Bjontegaard, G.: Calculation of average PSNR differences between RD-curves, Technical Report VCEG-M33, ITU-T SG16/Q6 (2001)
35. Gu, Z., Zheng, J., Ling, N., Zhang, P.: Fast intra prediction mode selection for intra depth map coding, ISO/IEC JTC1/SC29/WG11 (2013)
36. Gu, Z., Zheng, J., Ling, N., Zhang, P.: Fast intra SDC coding for 3D-HEVC intra coding, ISO/IECJTC1/SC29/WG11 (2014)