DECODING COMPLEXITY REDUCTION IN PROJECTION-BASED LIGHT-FIELD 3D DISPLAYS USING SELF-CONTAINED HEVC TILES

Alireza Zare[1], Péter Tamás Kovács[1], Alireza Aminlou[1, 2], Miska M. Hannuksela[2], Atanas Gotchev[1]

[1]Department of Signal Processing, Tampere University of Technology, Tampere, Finland
[2]Nokia Technologies, Tampere, Finland

ABSTRACT

The goal of this work is to provide a low complexity video decoding solution for High Efficiency Video Coding (HEVC) streams in applications where only a region of the video frames is needed to be decoded. This paper studies the problem of creating self-contained (i.e., independently decodable) partitions in the HEVC streams. Further, the requirements for building self-contained regions are described, and an encoder-side solution is proposed based on HEVC tile feature. A particular application of self-contained tiles targets the type of light-field 3D displays, which employ a dense set of optical engines to recreate the light field. Correspondingly, such 3D displays require a dense set of input views and therefore the partial decoding of bitstreams helps providing less complex and consequently real-time decoding and processing. The simulation results show a significant increase in decoding speed at the cost of a minor increase in storage capacity.

Index Terms **—** HEVC, light-field 3D displays, video partitioning, partial decoding, tile, slice, random access

1. INTRODUCTION

Light-field (LF) 3D displays [1] [2] [3] represent a major step forward in realistic 3D visualization. The new technology provides very high 3D quality experience through offering significantly higher resolution, higher brightness, smoother motion parallax, wider Field Of View (FOV), and larger depth range, when compared with typical auto/stereoscopic 3D displays [2]. LF displays can show imagery to multiple naked-eye freely moving observers, providing walk-around capability, much like holograms. This viewing freedom however comes at a high computational cost as it requires high amount of visual data to properly represent a scene from many angles. Interestingly, in the case when the display is driven by purely image-based content with no scene geometry available, the captured or rendered rays required to properly recreate sufficiently wide FOV with the angular resolution supported by the display are organized in number of views which can be hundreds. Delivering, processing and de/compressing this amount of views pose significant technical challenges.

Decoding a certain number of views of a given resolution on a single computer can become prohibitive, considering the video resolution and decoder speed. As it has been shown in [4], when processing light fields in a distributed system, access patterns in ray space are quite regular, some processing nodes do not need all views, moreover the necessary views are used only partially. This trait could be exploited to optimize decoding operation, thus enabling real-time operation of the system. While skipping the decoding of a single view is straightforward, partial decoding of video frames is not supported by existing video codecs, including HEVC, the current state-of-the-art video coding standard.

Partial decoding of a video frame requires random access to a portion of the bitstream associated with specific regions within the video frame [5] [6]. In the HEVC standard, similar to earlier standards, a video frame can be split into several partitions by using provided frame partitioning tools, which enable random access into portions of the bitstream with a specific granularity. Thus, the usage of the partitioning tools facilitates partial decoding, which is instrumental for feeding LF displays with live imagery.

To enable partial decoding, an encoder is configured to regularly divide video frames into a desired number of partitions with various possible arrangement in order to meet the requirements of the application under consideration. Afterward, an intermediate extractor is utilized to construct a conforming sub-bitstream representing the partitions necessary to be decoded. Hereby, the decoder is only provided with the coded elements corresponding to the pixel samples (i.e., the generated sub-bitstream by the extractor) needed by the LF rendering nodes, in the context of this paper. Consequently, this approach improves decoding speed by lowering computational burden of decoding. In the described scheme, however, it is assumed that partitions within a frame are independently decodable from each other. Moreover, a partition is independent from non-co-located partitions within the other frames. This kind of partitions, called self-contained, are essential in order to realize partial decoding. This research aims to enable self-contained partitions in the HEVC streams.

2. SELECTED ASPECTS OF THE HEVC STANDARD

The HEVC standard [7] adheres to block-based hybrid video coding paradigm. In the new standard, frame partitioning and block partitioning concepts are considerably improved in comparison with of those in Advanced Video Coding (AVC) standard. These improvements enable HEVC to be broadly flexible and optimized for a wide range of contents, applications and devices. In the following subsections, a brief overview of the HEVC standard is presented, with a particular focus on the relevant aspects to this work.

2.1 HEVC partitioning tools

In the HEVC standard, a video frame is partitioned into square-shaped regions called coding tree unit (CTU) which represents the basis of the coding process. The CTU in turn can be further subdivided into multiple coding units (CU) of different sizes based on a tree partitioning structure. The CU can then be split to form prediction units (PU), which are predicted using either intra-frame prediction or a motion compensation process.

In addition to the slice partitioning feature, as already known from the AVC standard, HEVC introduces a novel frame partitioning concept called tile [8]. The new afforded partitioning tool is primarily designed to facilitate parallel processing. When tiles/slices are enabled, an integer number of CTUs is aggregated into a tile/slice to form CTU-aligned frame partitioning. Each tile/slice is independently decodable from other tiles/slices within the same frame, where decoding refers to entropy, residual, and intra-frame prediction decoding. In fact, intra-frame prediction is

978-1-5090-3313-3/16/$31.00 ©2016 IEEE

limited within tile/slice boundaries and entropy coding state is re-initialized at the beginning of every tile/slice [9]. Due to these restrictions, both tile and slice come along with cost of increase in bitrate. However, this characteristic is advantageous for creating self-contained partitions.

Tiles appear to be more efficient than slices on a number of aspects. Tiles provide more flexibility to the partitioning, which brings any arbitrary partitioning arrangement. In contrast to slices, tiles incur fewer penalties of header data since tiles do not contain any headers. The first coded tile immediately follows the slice header to which it belongs. The starting points of the subsequent tiles' data in the bitstream, if any, can be explicitly signaled in the slice header. Unlike slices, tiles are always rectangular. Hence, tiles can be arranged in a more compact shape compared to the slices. This leads to less reduction of correlation between pixels within a tile, when compared to slice partitioning. Slices and tiles are alternatives that can be utilized for enabling video frame partial decoding. The two features can also be enabled together along with a restriction that tiles must include complete slices and vice versa.

2.2 HEVC intra and inter prediction schemes

Intra- and inter-prediction schemes have been demonstrated as the primary tools employed in the modern video coding standards. While intra-prediction scheme exploits spatial (i.e., within a video frame) redundancy, inter-prediction scheme drives a motion-compensated prediction (MCP) for a block of samples based on a translational motion model from reference frame(s). HEVC follows this basic idea in a more elaborated and flexible manner when compared with previous standards. This provides more efficient exploitation of redundancies among video frames and thus improving compression efficiency. However, it brings complexity to other aspects of video coding including random access to bitstream, where eliminating or removing the prediction dependency is advantageous. For example, in applications where partial decoding of video frames is desired, both intra- and inter-prediction dependencies among constructed partitions have to be removed, in addition to other kinds of dependencies.

As discussed in Subsection 2.1, using tiles and slices restricts intra-frame prediction within tile/slice boundaries. However, in inter-frame prediction a block in a tile/slice can be predicted from whole the reference frame with disregarding tile/slice boundaries. In other words, in motion compensation process motion vectors of a block in the current tile/slice can potentially point to any region out of the co-located tile/slice boundaries in the reference frame. Hence, this issue prevents a sequences of tiles/slices from being independently decodable and further applicable in partial decoding applications.

In the HEVC inter-prediction scheme, two new techniques so-called advanced motion vector prediction (AMVP) and inter-prediction block merging are introduced to efficiently single motion information. These techniques provide efficient tools for exploiting correlation among motion data of neighboring blocks. They are based on the fact that motion vectors of spatial and temporal neighboring blocks are well correlated, since they are likely to belong to the same moving object with similar motion. For both AMVP and merge methods, a list of spatio-temporal candidates is constructed. Afterward, the index of the best candidate to be used for inferring the motion information of the current block is signaled. The filling of the candidate list is standardized and candidates' order is fixed such that the signaled index would refer to the same candidate to the list at the decoder side. Figure 1 depicts the most suitable spatio-temporal candidates for the merge and AMVP techniques [7]. The current and colocated PUs belong to the current and the reference frames, respectively.

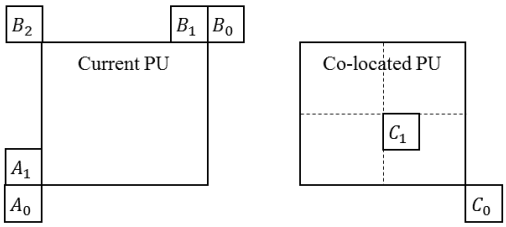

Figure 1. Spatio-temporal positions of merge and AMVP candidates

In the derivation of spatial candidates, a candidate is considered as not available if the associated PU belongs to another slice/tile, or is intra coded. The temporal candidate is selected between C_0 and C_1 positions. If PU at position C_0 is not available, beyond the current CTU, or intra coded, position C_1 is selected [7]. As a consequence, in both motion data signaling methods, spatial and temporal candidates are selected such that they belong to the current tile/slice and the co-located one in the reference frames, respectively. The way of constructing the candidate list facilitates the constructed partitions to be considered self-contained. However, both tile and slice partitioning tools still suffer from the fact that motion vectors can freely point to any predictor beyond the current and co-located tile/slice in the reference frames. Although, this behavior improves compression efficiency, it is problematic in the context of creating self-contained partitions, since it may lead to breaking tiles/slices boundaries. To remedy this drawbacks of inter-prediction scheme, some restrictions in the encoder side has to be imposed, which are described in the next section.

2.3 HEVC bitstream syntax

This section provides a brief overview of the HEVC stream syntax, particularly the related part to this work. Dropping coded elements related to unnecessary partitions from bitstream requires knowledge of how coded elements are arranged in the bitstream. HEVC inherits hierarchical Network Abstraction Layer (NAL) unit based bitstream structure from AVC and uses parameter set concepts similarly to AVC. NAL units are byte aligned and each NAL unit consists of a NAL unit payload and a two-byte NAL unit header identifying the type of payload including Video Coding Layer (VCL) or non-VCL NAL unit.

Each NAL unit in the bitstream is separated by some specific start codes or framing (e.g. by the container file format or communication protocol). Four bytes equal to 0x00000001 and three bytes equal to 0x000001, are designated to separate coded frames and slices, respectively. Each coded frame together with the associated non-VCL NAL units is called an HEVC access unit. From this perspective, an HEVC stream can be seen as series of access units. In the case of tile partitioning, the entry points of tiles can be signaled in the slice header or provided as metadata in the container file format. In contrast to slices, no start codes or framing are designated for tiles. Accessing each tile requires reading its entry point from the slice header in which the tile belongs or from the metadata in the container file.

3. SELF-CONTAINED TILES

3.1 Motion estimation restriction

The independency of self-contained tiles must be guaranteed at the encoder side such that each constructed tile is independently decodable from other tiles within the same frame and also non-co-located tiles in reference frames, at the decoder side. As described in Subsection 2.2, an HEVC encoder has to be modified in order to provide self-contained tiles. Particularly, some restrictions have to be imposed upon motion estimation process

such that it only utilizes the area within the co-located tile as reference area. The search range is initially constrained to lie within the co-located tile. Each time that a predictor is estimated, its associated motion vector is examined to be within the restricted search range. The search range is restricted within the co-located tile by considering interpolation filter tap size, in case of sub-sample prediction, and the current PU size. No restriction is imposed on the temporal merge and AMVP candidates since the selection is performed in such way that tile boundaries are not broken. However, their motion vectors are examined to ensure that they point to the restricted area. In case a motion vector cross tile boundaries, the associated candidate is signaled as unavailable.

3.2 Self-contained tile extractor

A tile-based extractor was implemented on high syntax level to generate an HEVC conforming sub-bitstream from a bitstream containing the whole video sequence. The NAL units related to desired tiles are extracted from bitstream and copied to the output sub-bitstream. Moreover, in order to provide an HEVC conforming bitstream altering some parameters in the parameter sets are required. These parameters include slice segment address, dimension of the output video sequence, number of tile rows and columns, and tile enabled flag. The latter is disabled in case the output sub-bitstream contains only one tile. The operation results in a smaller bitstream which can be decoded using a standard decoder.

3.3 Combined tile-slice partitioning

Partitioning of the video frames into regions can be achieved by utilizing only tiles. However, in this study the partitioning is performed using a combination of tile and slice. The reasons are to allow partitioning of video frames to vertical stripes, and low-complexity implementation of partial decoding operation. According to the HEVC standard, the end of a slice is indicated using the end_of_slice_flag which is the last encoded element for each CTU. It is set to one if the current CTU is the last CTU in the slice. In case of using tile alone as partitioning tool, only for the last CTU of the last (i.e. most right) tile within frame the flag is set. Assume that in a light-field rendering node, only the pixel samples corresponding to the first tile is requested. Therefore, the extractor generates a sub-bitstream containing the first tile of each frame, in which the end_of_slice_flag equals to 0. The generated sub-bitstream is not decodable by a standard HEVC decoder, as the decoder does not meet the end of slice condition while decoding the last CTU within the frame in the output sub-bitstream. The flag is entropy coded and cannot be modified without entropy decoding the slice data. The proposed partitioning helps to avoid such situation as the last CTU within overlapped slices and tiles are the same. Hence, the flag is set for each tile. However, it comes along with a cost of increase in bitrate. Table 1 shows that the bitrate increment caused by slice header is less than 1% for the experimented video sequences. The amount of overhead is such small that can be negligible.

4. EXPERIMENTAL RESULTS

The proposed encoder modifications were implemented in the HEVC reference software HM version 16.7 [10] to evaluate the compression efficiency. The self-contained tile extractor was used to extract the required sub-bitstream, and the standard decoder was used to examine HEVC conformance and analyze decoding complexity. The reported experiments are conducted on test sequences of different contents from low motion to high motion. The simulation were performed using random access configuration with 89 frames per sequence. The quantization parameters (QPs) are selected in the range of 22-34 with delta QP equal to 4. The reported results include compression efficiency in terms of Bjøntegaard Delta-rate (BD-rate) criterion [11] and decoding speed in unit of frames per second. The qualitative performance is not discussed since the proposed modifications has almost no effect on the PSNR values.

4.1 Partitioning arrangement

In this experiment, the encoder is configured to subdivide video frames into four vertical tiles in a regular manner, such that each tile occupies the same spatial region over all frames. This kind of vertical partitioning is practical in the LF conversion process employed in the LF displays. Each tile covers about 25% of the whole frame and contains exactly one slice which its boundaries match the tile's boundaries. Figure 2.a illustrates the described partitioning arrangement for the Balloons video sequence.

4.2 Compression performance

As discussed in Subsection 2.1, the usage of tiles and slices comes along with compression efficiency penalty. The finer the partitioning is, the higher the compression loss occurs. Table 1 represents the compression loss for cases in which video frames are partitioned using only tiles, a combination of normal tiles and slices and a combination of self-contained tiles and slices, for the partitioning scheme described in the previous section. The BD-rate results show that breaking of intra prediction and re-initialization of entropy coding engine (i.e., only tiles used) cause 1.5% reduction in compression efficiency in average, when compared with no partitioning case. Additionally, the usage of slice arises with 0.6% (= 2.10% – 1.5%, in Table 1) higher compression loss caused by slice header overhead. Moreover, the results indicate that the proposed modifications into the encoder, in order to provide self-contained tiles, drop compression efficiency by 4.43% (= 6.53% – 2.10%, in Table 1). The latter loss caused by restricting motion vectors within co-located tile boundaries in the

a) Partitioning arrangement

b) One tile extraction

c) Two tiles extraction

Figure 2. Frame partitioning and tile-based extraction

Table 1. BD-Rate overhead (%)

Sequences	Tiles	Normal tiles & slices	Self-contained tiles & slices
Balloons	1.46	2.39	6.16
Pantomime	0.79	1.17	4.02
Shark	3.09	3.78	10.11
BBB_Flower	2.00	2.78	2.09
Kimono	0.14	0.40	10.26
Avg.	1.50	2.10	6.53

Table 2. Average decoding speed (frames per second)

QP	1 tile	2 tiles	3 tiles	Whole frame
22	64.04	30.04	19.81	13.18
26	77.44	37.98	25.09	16.77
30	84.73	42.06	28.20	18.61
34	87.34	44.59	29.34	19.75
Avg.	78.39	38.67	25.61	17.08

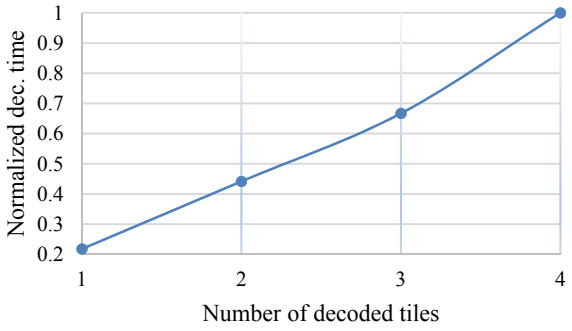

Figure 3. Normalized decoding time

reference frames, which results in temporal correlation reduction in the inter-prediction scheme. Overall, the proposed approach introduces 6.53% loss in compression efficiency for the employed partitioning scheme over the test videos.

4.3 Complexity and latency reduction

In order to provide some insight in the decoder speed increment, four different scenarios were tested. They include the cases where a LF rendering node requests the area corresponding to: one tile, two tiles, three tiles, and four tiles (i.e. the whole frame). In the display side, only the requested tile(s) are received and decoded.

Figure 2b and Figure 2c illustrate the cases where one tile and two tiles are decoded, respectively. In Table 2, average decoding speed, over the test videos, for the test scenarios with different QPs is represented. It shows a significant increase in decoding speed by decoding only the desired region. Compared to decoding the whole frame area, the decoding speed increases 4.6, 2.3, and 1.5 times when decoding one, two and three tiles, respectively. In Figure 3, the normalized decoding time is shown, with respect to decoding time of the whole frame. It can be seen that the decoding time has almost a linear relation with the number of decoded tiles.

5. CONCLUSION

This paper has provided an elegant solution for enabling partial decoding of an HEVC stream in applications where only a region of the video frames is decoded. It has elaborated the creation of independently decodable partitions, so-called self-contained partitions, in the HEVC streams using tiles and slices. The paper has formulated the requirements that have to be imposed on the HEVC standard HM encoders in order to enable self-contained partitions. The performance of the proposed self-contained tiles was evaluated in the context of light-field 3D displays, in which partial decoding helps in enabling real-time operation. Naturally, the proposed approach is not limited to the light-field 3D displays. It can be utilized in any applications (e.g. ROI applications) where video frames can be reasonably split into separate partitions and only a sequence of the partitions need to be decoded.

The results indicate that the usage of partitioning tools and the proposed encoder modifications yield a small negligible compression penalty. However, the proposed approach makes the decoding operation very efficient and low complex by reducing computational burden of decoding. It significantly increases decoding speed and thus helps enabling real-time processing.

6. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the PROLIGHT-IAPP Marie Curie Action of the People programme of the European Union's Seventh Framework Programme, REA grant agreement 32449. The work was further developed in Nokia Technologies in Tampere.

7. REFERENCES

[1] T. Balogh, P. T. Kovács and Z. Megyesi, "Holovizio 3D display system," in *proc. IMMERSCOM 2007*, 2007.

[2] W. G. and et al., "Tensor displays: compressive light field synthesis using multilayer displays with directional backlighting," in *proc. SIGGRAPH 2012*, 2012.

[3] K. M. and et al., "Glasses-free 200-view 3D video system for highly realistic communication," in *proc. Digital Holography and Three-Dimensional Imaging, OSA Technical Digest, paper DM2A.1*, 2013.

[4] P. T. Kovács, Z. Nagy, A. Barsi, V. K. Adhikarla and R. Bregović, "Overview of the applicability of H.264/MVC for real-time light-field applications," in *proc. 3DTV-CON 2014*, Budapest, 2014.

[5] M. M. Hannuksela, Y.-K. Wang and M. Gabbouj, "Isolated regions in video coding," *IEEE Transactions on Multimedia,* vol. 6, p. 259–267, 2004.

[6] A. Zare, P. T. Kovács and G. Atanas, "Self-Contained Slices in H.264 for Partial Video Decoding Targeting 3D Light-Field Displays," in *Proc. 3DTV Conference*, Lisbon, 2015.

[7] C. Rosewarne, B. Bross, M. Naccari, K. Sharman and G. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16), Document JCTVC-U1002," Warsaw, Jun. 2015.

[8] K. M. Misra, C. A. Segall, M. Horowitz, S. Xu, A. Fuldseth and M. Zhou, "An overview of tiles," *IEEE Journal of Selected Topics in Sig-nal Processing,* vol. 7 , no. 6, pp. 969-977, Dec. 2013.

[9] H. Schwarz, T. Schierl and D. Marpe, "Block Structures and Parallelism Features in HEVC," in *High Efficiency Video Coding (HEVC): Algorithms and Architectures*, Springer, 2014, pp. 49-90.

[10] "High Efficiency Video Coding (HEVC)," Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, [Online]. Available: https://hevc.hhi.fraunhofer.de/.

[11] G. Bjøntegard, "Calculation of average psnr differences between rd-curves, document VCEG-M33," Austin, 2001.