Implementing Viewport Tile Extractor for Viewport-Adaptive 360-Degree Video Tiled Streaming

Jong-Beom Jeong
Department of Computer Education
Sungkyunkwan University (SKKU)
Seoul, Republic of Korea
uof4949@skku.edu

Soonbin Lee
Department of Computer Education
Sungkyunkwan University (SKKU)
Seoul, Republic of Korea
soonbinlee@skku.edu

Inae Kim
Department of Computer Education
Sungkyunkwan University (SKKU)
Seoul, Republic of Korea
inaelk@skku.edu

Eun-Seok Ryu
Department of Computer Education
Sungkyunkwan University (SKKU)
Seoul, Republic of Korea
esryu@skku.edu

***Abstract*—Because 360-degree video streaming has become significantly popular in the field of virtual reality, the viewport-adaptive tiled streaming technology for 360-degree video is emerging. This paper presents a viewport tile extractor (VTE) that is implemented on high-efficiency video coding (HEVC). The VTE extracts multiple tiles that represent the viewport of a user and merges them into one bitstream. The proposed system transmits the bitstream of high-quality tiles and the low-quality video bitstream of entire area to reduce both latency and bandwidth. The proposed method shows more than 16.98% of bjontegaard delta rate saving in terms of the luma peak signal-to-noise ratio, compared with the HEVC-compliant streaming method. Additionally, compared with the existing tiled streaming method, it achieves 66.16% and 69.79% saving of decoding memory and time consumption, respectively.**

***Index Terms*—Virtual reality, HEVC, MCTS, Viewport-adaptive streaming, 360-degree video**

I. Introduction

Recently, virtual reality (VR) has become considerably popular, and various head-mounted display (HMD) devices are being developed to meet the consequent requirements. Additionally, the demand for realistic 360-degree scene representation on HMDs is increasing. To reduce the motion sickness (nausea) and increase the quality of experience (QoE) of users, high-quality low-latency video streaming is required. Champel et al. [1] reported that a video that has the resolution of 12K, frame rate of 90 frames per second (fps), and motion-to-photon latency (MTP) of 20ms is needed to meet the requirements.

Because the requirements of [1] are challenging to meet, several approaches were proposed to overcome the limitations of the previous researches. For example, in high-efficiency video coding (HEVC), motion-constrained tile set (MCTS) was proposed [2]. MCTS limits the temporal inter prediction at the rectangular tile boundaries and removes the correlation between the tiles. Using an MCTS-based tile extractor, each tile can be extracted from the entire video bitstream and then decoded. Because of the MCTS-based approach, viewport-adaptive tiled streaming methods have progressed [3] [4]. Because a user watches a part of the entire 360-degree video via HMD, transmitting only the field of view (FoV) of the user can reduce the bandwidth while providing high-quality video to the user. However, the existing reference HEVC tile extractor can extract only one tile at once. Although tiled streaming reduces bandwidth, the existing method generates bitstreams of multiple tiles and thus requires several decoders, thereby increasing the decoding time and computation power requirements.

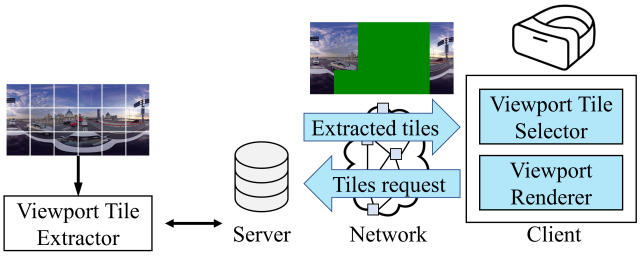


Fig. 1: Overview of the viewport-adaptive tiled streaming

To overcome the aforementioned challenge, this paper proposes a viewport tile extractor (VTE) for realizing viewport-adaptive tiled streaming. Figure 1 shows an overview of the viewport-adaptive tiled streaming. The proposed system encodes 360-degree videos and generates the corresponding MCTS bitstreams. At the client side, a viewport tile selector detects viewport area tiles, and the viewport tile indices are given to the VTE. Subsequently, the VTE extracts multiple tiles and merges them into a single bitstream irrespective of

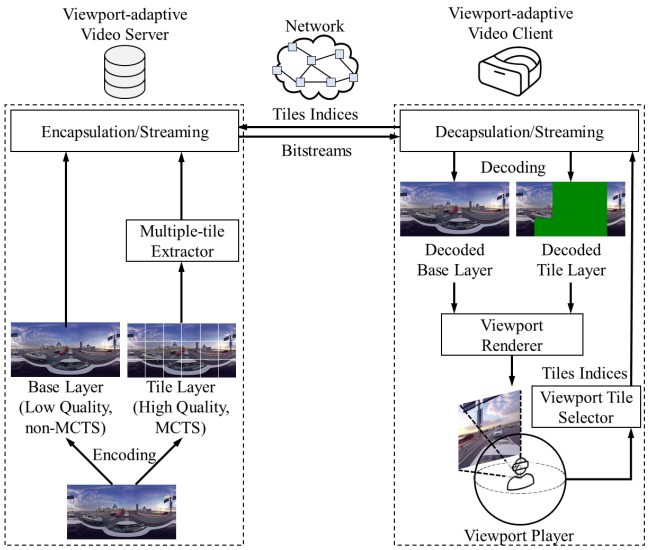


Fig. 2: Two-layer viewport-adaptive 360-degree video tiled streaming

the number of the tiles. After the bitstream is transmitted, the client decodes the bitstreams. Finally, a viewport renderer generates the viewport and the video is displayed through HMD.

The remainder of this paper is arranged as follows. Section II explains the related work, and Section III describes the viewport-adaptive VTE schemes. The experimental settings, results, and analyses are explained in Section IV. Finally, Section V draws the conclusions and presents insights into the future work.

II. Related Work

In HEVC, tile was proposed to support parallel processing. A video can be divided into rectangular tiles, each of which can be simultaneously decoded. MCTS [2] was, therefore, proposed to use tiles at the bitstream level. It limits the temporal inter prediction at the tile boundaries and removes the correlation between the tiles. Therefore, if a video was encoded using an MCTS encoder, each tile can be extracted from a bitstream, and it can be decoded using the HEVC standard decoder. In the HEVC test model (HM) version 16.20 [5], the implementations of MCTS and tile extractor are included. The extractor can generate one bitstream for one tile. In 360-degree video streaming, only a part of the entire 360-degree video is displayed on the HMD. Therefore, to save the bandwidth, only a part of the 360-degree video can be streamed. The tiles that belong to the viewport of user can be extracted from the MCTS bitstream and then transmitted. Zare et al. [6] showed the tiled streaming performance of VR. They stated that the bitrate gain increased upon reducing the tile size.

III. Viewport-Adaptive 360-Degree Video Tiled Streaming

This section explains the proposed two-layer viewport-adaptive 360-degree video tiled streaming based on the head movement of user, as shown in Figure 2. The processes of the proposed system are as follows. A 360-degree video is encoded using a HEVC encoder, and the server generates a high-quality tile layer and low-quality base layer. The base layer covers the entire 360-degree video, and MCTS is not applied. However, the tile layer contains the viewport-area tiles, and it is encoded using MCTS. The viewport tile selector detects the viewport area and provides the information of the tiles to the VTE. Subsequently, the VTE extracts tiles from the high-quality tile layer bitstream and generates single bitstream. At the client side, only two decoders are required to decode the base and tile layers, while the existing tile extractor based streaming requires many decoders. Therefore, the proposed system can provide high-quality, low-latency video streaming by simulcasting the low-quality base layer and high-quality tile layer. Although the user may turn his/her head, the low-quality area is displayed for only a short period and the transmitted high-quality viewport tiles are decoded and rendered. The remainder of this section introduces the VTE scheme and its advantages.

A. Viewport Tile Extractor

As previously mentioned in Section II, if the number of tiles in a picture increases, the bandwidth is reduced because if the tile size become smaller, unnecessary areas are excluded. However, in a practical system, a small grid partitioning generates more number of bitstreams of viewport tiles, thereby increasing the computation power requirements and processing time. Because the existing reference HEVC tile extractor generates one bitstream for each tile, many decoders are required at the client side to reduce the processing time. If a device has limited resources, a small number of decoders can be executed. Therefore, the existing tiled streaming has to strike a trade-off between bandwidth and resource utilization.

To overcome the trade-off, VTE generates single bitstream that contains multiple tiles. Figure 3 shows the functional flow chart of the VTE scheme. First, the indices of the viewport tiles, and the bitstream are given. Because an HEVC bitstream comprises a network abstraction layer (NAL) unit, the VTE parses each NAL unit. If MCTS is applied to the bitstream, it contains video parameter sets (VPS), sequence parameter sets (SPS), picture parameter sets (PPS), extraction information sets (EIS) supplemental enhancement information (SEI) message, and slice. Notably, VPS, SPS, and PPS contain the information of the bitstream, and the VTE parses the necessary information from the parameter sets. Because the parameter sets are parsed in the decoder loop, obtaining the parameter sets directly from the bitstream is not possible using a standard decoder. Therefore, upon enabling the MCTS, the encoder encodes the parameter sets of each tile into EIS SEI messages. The VTE then obtains the parameter sets from the EIS SEI messages. On the basis of the parsed information, the VTE

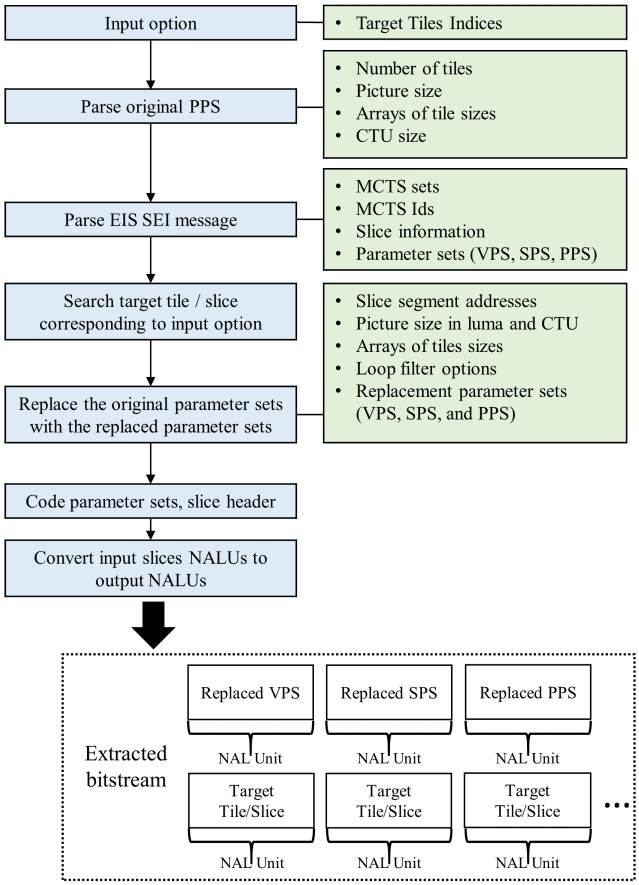
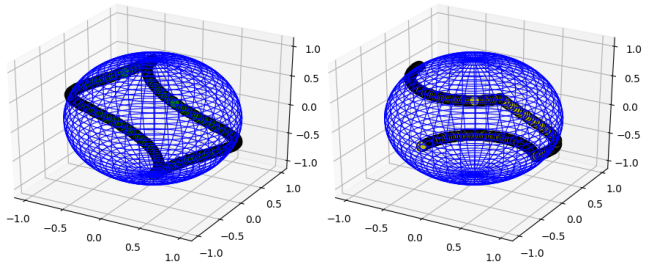


Fig. 3: Functional flow chart of the VTE scheme

determines the slices that have the target tiles. Meanwhile, the obtained parameter sets contain the information of the individual tiles. They can be used for a single tile bitstream. However, the VTE generates a bitstream that has the same picture size as that of the input bitstream. Accordingly, the VTE replaces some of the options of obtained parameter sets with the options of replaced parameter sets including slice segment addresses, picture size, arrays of the tile sizes, and loop-filter options. After the replacement, the parameter sets and the slice headers are coded. The input NAL units, which have the target slices, are converted into output NAL units, which are then inserted into the output bitstream. Finally, the VTE generates a bitstream that has multiple target tiles.

The followings are the advantages of the VTE:

- Unlike the previous extractor explained in Section II, the VTE generates only one bitstream irrespective of the number of target tiles. This method requires only one decoder at the client side, along with few memory resources and less processing time.
- The VTE is simple, generic, and compatible with the existing HEVC standard. Additionally, no changes are required to the decoder.



(a) (b)

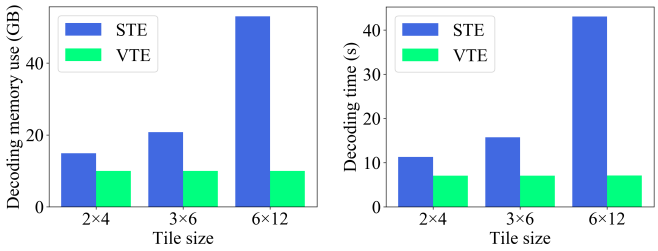
Fig. 4: Traces of the viewport movements in (a) option 1, (b) option 2

IV. Experimental Results

A. Experimental Conditions

This section introduces the experimental conditions of the proposed method. HM version 16.20 [5] was used for implementing the proposed method. For viewport generation, 360lib version 5.1-dev [7] was used. To evaluate the performance of the proposed method, metrics, namely, peak signal-to-noise ratio (PSNR), video multimethod assessment fusion (VMAF) [8], multi-scale structural similarity (MS-SSIM) [9], and immersive video PSNR (IV-PSNR) [10], were used. These metrics were analyzed using the bjontegaard delta rate (BD-rate) to show the bitrate savings against the video quality. The experiments were conducted on the basis of the JVET common test condition (CTC) for 360-degree videos [11]. According to CTC, four test sequences were used: AerialCity, DrivingInCity, DrivingInCountry, and PoleVault_le. These sequences are 4K video clips of length 10 seconds with 300 frames at 30 fps. Three grid partitionings were used: 2×4, 3×6, and 6×12. THe HEVC random access (RA) main profile was used to encode the test sequences. For quantization parameters (QPs), 22, 27, 32, and 37 were used to encode the tile layer, and 42 to encode the base layer. Considering the streaming scenario and main profile, the videos were divided into 32 frame chunks. For all the 32 frames, the sum of the viewport tiles was extracted. We used the viewport movement scenarios of Tomohiro et al. [12], which proposes four dynamic viewport movement scenarios, and options 1 and 2 are recommended. Therefore, we used option 1 and 2 for viewport generation. Figure 4 shows the traces of the viewport movements of options 1 and 2.

The server used for this experiment had two Intel Xeon E5-2687w v4 CPUs (24 cores and 48 threads), 128 GB of memory, GTX 1080 Ti, and Ubuntu 18.04 installed. At the client side, there existed Windows 10 desktop that had Intel i7-7700k processor (4 cores and 8 threads), 16GB of memory, and the GTX 1080 Ti graphics card. To measure the decoding time and the memory consumption at the client side, hevc_nvenc (NVIDIA NVENC HEVC encoder) ffmpeg decoder was used.



(a) (b)

Fig. 5: Performance comparison between the STE and VTE in terms of (a) decoding memory use, (b) decoding time

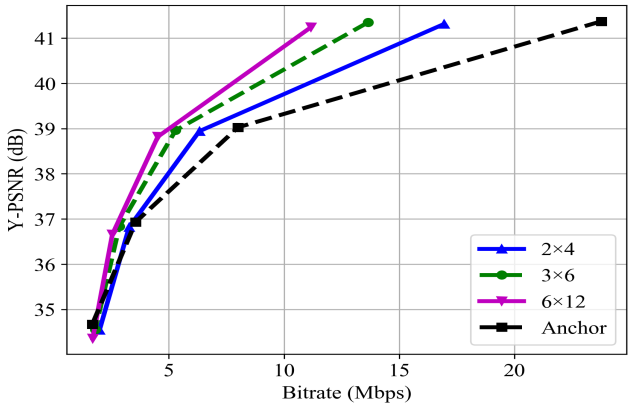


Fig. 6: Y-PSNR RD-curve: HEVC anchor versus 2×4-, 3×6-, 6×12-tiled streamings

B. Analysis of the Results

This section describes and analyzes the experimental results of the proposed tiled streaming system. To evaluate the decoding performance, we compared the decoding results of a single tile extractor (STE) from HM 16.20 with those of the VTE. Figure 5 shows the decoding memories and time consumptions of the two methods. Upon dividing the videos into 6×12-grid tiles, the STE consumed 52.99 GB of memory and 43.04 seconds for decoding the viewport tiles of 300 frames, while the VTE required only 10.00 GB of memory and 7.08 seconds. Moreover, the VTE showed similar decoding resource consumption for three grid partitionings, meaning that the dense grid partitioning can be used to reduce the bandwidth without considering the decoding resources.

Table I compares the BD-rate of the proposed tiled streaming method with that of the non-tiled streaming method. In average, the proposed method showed the BD-rate savings of 16.98%, 6.45%, 11.13%, and 15.77% for Y-PSNR, VMAF, MS-SSIM, and IV-PSNR, respectively. Compared with transmitting only high-quality tiles, the proposed method required higher bitrate. However, the proposed method offered low-latency, which is significantly important in the real streaming system. Figure 6 shows the rate-distortion (RD) curve of the HEVC anchor and proposed tiled streaming method. Among the three grid partitionings, the 6×12-grid one achieved the highest BD-rate saving. As shown in Figure 5, the VTE can be used to realize dense-grid tiled streaming. Figure 7 shows the viewports generated by the HEVC anchor, 2×4, 3×6, and 6×12-grid tiled streaming. As shown in the figure, in the HEVC anchor viewport, there exist several noticeable visual artifacts: the area of window, car door, and street lamp. However, these artifacts are not shown in the 6×12-grid tiled streaming. The proposed method using 6×12-grid tiling, which is shown in Figure 7, requires 2.58 Mbps for the tile layer and 0.69 Mbps for the base layer. Although the proposed method requires a slightly higher bitrate compared with that of the HEVC anchor in Figure 7, the required bitrate can be lowered upon using a higher QP for the base layer.

TABLE I: Streaming bitrate performance of the proposed tiled streaming method compared to the non-tiled streaming method (BD-rate (%))

Tiling	Y-PSNR	VMAF	MS-SSIM	IV-PSNR
2×4	-6.05	2.95	-1.73	-5.14
3×6	-19.93	-8.36	-12.90	-17.86
6×12	-24.97	-13.96	-18.77	-24.32
Average	**-16.98**	**-6.45**	**-11.13**	**-15.77**

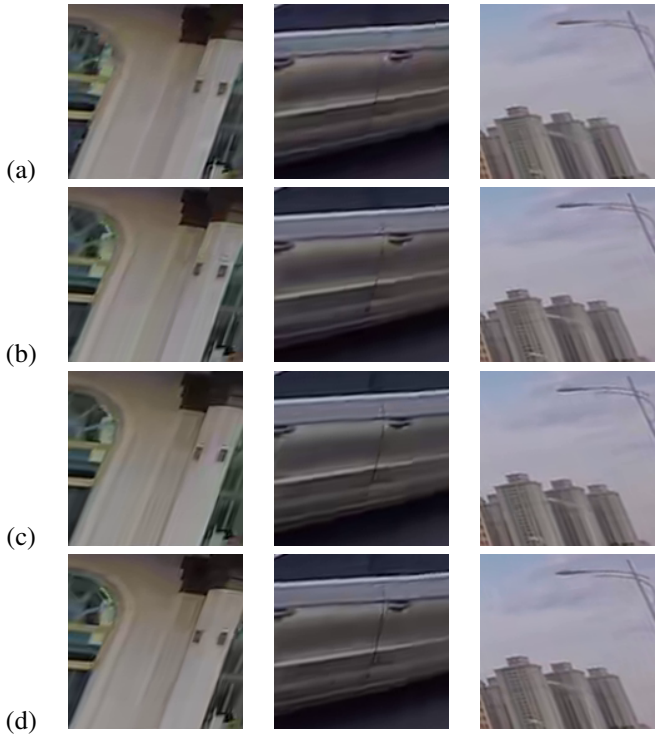


Fig. 7: Generated viewport comparison with enlarged noticeable sections in DrivingInCity: (a) anchor, 3.21Mbps@39.16dB, (b) 2×4 tiling, 3.24Mbps@40.26dB, (c) 3×6 tiling, 3.32Mbps@40.66dB, (d) 6×12 tiling, 3.28Mbps@40.55dB

V. Conclusion

This paper proposed a two-layer viewport-adaptive 360-degree video tiled streaming. Specifically, the server generates

a low-quality base layer and high-quality tile layer to reduce the streaming delay. The base layer contains a full picture, while the tile layer includes viewport tiles. The proposed VTE generates single bitstream that contains multiple tiles, thereby decreasing the decoding memory use by 66.16% and decoding time by 69.79%, as compared to the STE. Overall, it showed a BD-rate saving of 16.98% for Y-PSNR. The proposed method is simple, efficient, and compatible with the existing HEVC devices. In future, extensive experiments for determining the optimal base layer QP and grid partitioning will be conducted.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2019R1A2C1010476).

REFERENCES

[1] M.-L. Champel, T. Stockhammer, T. Fautier, E. Thomas, and R. Koenen, "Quality requirements for vr." 116th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG116/m39532., 2016.

[2] R. Skupin, Y. Sanchez, K. Sühring, T. Schierl, E.-S. Ryu, and J. Son, "Temporal mcts coding constraints implementation." 122th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG 122/m42423., 2018.

[3] J. Son, D. Jang, and E.-S. Ryu, "Implementing 360 video tiled streaming system," in *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, 2018, pp. 521–524.

[4] J. Son and E.-S. Ryu, "Tile-based 360-degree video streaming for mobile virtual reality in cyber physical system," *Computers & Electrical Engineering*, vol. 72, pp. 361–368, 2018.

[5] Hevc test model (hm) 16.20. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.20

[6] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 601–605.

[7] 360lib 5.1-dev. [Online]. Available: https://jvet.hhi.fraunhofer.de/svn/svn_360Lib/branches/360Lib-5.1-dev/

[8] C. G. Bampis, A. C. Bovik, and Z. Li, "A simple prediction fusion improves data-driven full-reference video quality assessment models," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 298–302.

[9] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[10] A. Dziembowski, "Software manual of iv-psnr for immersive video." 128th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG127/n18709, 2019.

[11] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, "Jvet common test conditions and evaluation procedures for 360° video." 118th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG2017/n16891, 2017.

[12] T. Ikai, Y. Yasugi, and T. Aono, "Ahg8: Dynamic viewport generation for 360° video evaluation." 127th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG2017/m39669, 2017.