DWS-BEAM: Decoder-Wise Subpicture Bitstream Extracting and Merging for MPEG Immersive Video

Jong-Beom Jeong*, Soonbin Lee*, Eun-Seok Ryu†
*Department of Computer Science Education, Sungkyunkwan University (SKKU), Seoul, Republic of Korea
†Department of Computer Education, Sungkyunkwan University (SKKU), Seoul, Republic of Korea
E-mail: {uof4949, soonbinlee, esryu}@skku.edu

***Abstract*—With the new immersive video coding standard MPEG immersive video (MIV) and versatile video coding (VVC), six degrees of freedom (6DoF) virtual reality (VR) streaming technology is emerging for both computer-generated and natural content videos. This paper addresses the decoder-wise subpicture bitstream extracting and merging (DWS-BEAM) method for MIV and proposes two main ideas: (i) a selective streaming-aware subpicture allocation method using a motion-constrained tile set (MCTS), (ii) a decoder-wise subpicture extracting and merging method for single-pass decoding. In the experiments using the VVC test model (VTM), the proposed method shows 1.23% BD-rate saving for immersive video PSNR (IV-PSNR) and 15.78% decoding runtime saving compared to the VTM anchor. Moreover, while the MIV test model requires four decoders, the proposed method only requires one decoder.**

***Index Terms*—Virtual reality, MIV, 6DoF, VVC, MCTS, Tile, Subpicture, Metaverse**

I. Introduction

Virtual reality (VR) 360-degree video streaming is a well-studied research area and is used in many fields, such as immersive education, autonomous driving, and entertainment. In contrast to traditional 2-D plane videos, only some of the entire 360-degree video is rendered to the head-mounted display (HMD), which results in high-resolution and high-quality video streaming. To address high bandwidth and the computational complexity with limited resources, several viewport-aware tiled streaming researches have been conducted [1]–[5]. Recently, the MPEG-immersive (MPEG-I) group has been studied the six degrees of freedom (6DoF) compression, which allows a user to move freely in a VR environment using multi-view videos captured by real or virtual cameras. Therefore, the MPEG immersive video (MIV) coding standard has been established [6], and the test model for immersive video (TMIV) was developed as a MIV reference software. The internal processes of the TMIV are as follows. The input views are divided into two groups: basic views (BVs) and additional views (AVs). BVs contain most of the information from multi-view videos, and views that do not belong in the BVs are the AVs. After BV allocation, two main steps of the TMIV are conducted: pruning and packing. In the pruning step, the redundancy between AVs is removed in the pixel domain. After pruning, the packing module finds the remaining information and segments it in the form of rectangular patches. These patches are then merged into atlases and encoded using 2-D video codecs such as high-efficiency video coding (HEVC) [7] or versatile video coding (VVC) [8]. The latest TMIV obtains texture (colour) and geometry (depth) views and outputs four atlases, while the geometry atlases are downsampled by a factor of 2×2.

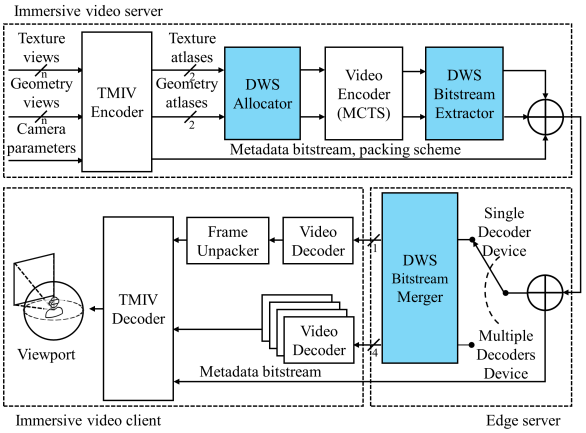

Fig. 1: System architecture of the proposed DWS-BEAM for MPEG immersive video

This paper proposes a decoder-wise subpicture bitstream extracting and merging (DWS-BEAM) method for MIV, as shown in Figure 1. The proposed method provides fine-grained subpicture merging schemes which can be used in both high- and low-end devices. After the TMIV encoding, the four atlases are divided into tiles/subpictures by the DWS allocator with consideration for two factors: division of the BV and AV patches, and merging into a single picture. By using a motion-constrained tile set (MCTS), each tile/subpicture can be extracted and merged in the compressed domain. The atlases are encoded by a 2-D video encoder and divided into subpictures. The DWS bitstream extractor extracts each subpicture and transmits to the edge server. The edge server receives the client's decoder status. If the client has a single decoder, the DWS bitstream merger merges the subpicture bitstreams into a single bitstream. The merged bitstream is then transmitted to the client and decoded, and then the frame unpacker reconstructs the four atlases. Otherwise, the bitstreams are merged into four bitstreams, as provided by the existing TMIV. Finally, the user's viewport is rendered.

The remainder of this paper is organised as follows. Section II explains the related work, and Section III describes the

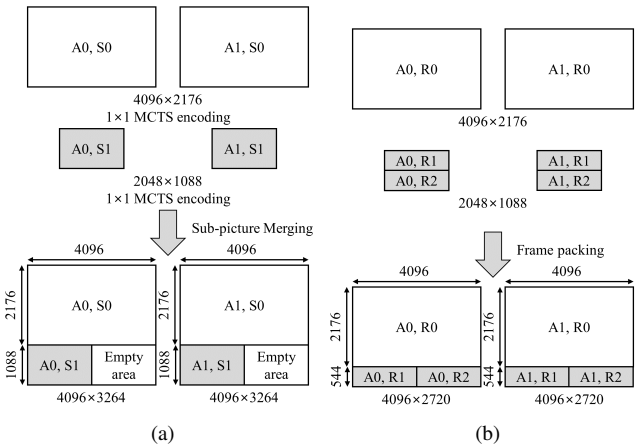

Fig. 2: Frame packing methods by (a) m54274 [11], (b) m56827 [13]. Abbreviations: A, atlas; S, subpicture; R, region

DWS-BEAM for MIV. The experimental setting and results are presented in Section IV. Finally, Section V presents the conclusion.

II. Related Work

Although the TMIV reduces the number of multi-view videos and generates four atlases, in legacy devices which have a single decoder, four bitstreams still present a challenge in terms of decoding time and bandwidth [9]. Therefore, frame packing, which packs both texture and geometry data into a single video, was proposed and studied in the MIV core experiment (CE) 1.3 [10]. In this paper, two researches are introduced: one from Nokia and the other from Intel. Nokia submitted document m54274 and proposed the VVC subpicture-based frame packing method, as shown in Figure 2a [11]. Note that the white box indicates the texture atlas, while the grey box represents the geometry atlas. In this method, each atlas is encoded as a subpicture using MCTS, and one texture and geometry subpicture are aligned horizontally and merged by VVC subpicture merging software [12]. Therefore, two decoders are needed, but there are empty areas in the merged picture. Intel submitted document m56827 to propose the frame packing implementation in the TMIV, which deploys a pixel-domain approach, as shown in Figure 2b [13]. Each texture atlas is regarded as one region, and each geometry atlas is divided horizontally to prevent the empty areas. At the bottom of the texture atlas, the divided geometry atlas is packed in the pixel domain; therefore, two decoders are needed.

III. Decoder-Wise Subpicture Bitstream Extracting And Merging for MIV (DWS-BEAM)

This section explains the proposed DWS-BEAM method for MIV. The decoder-wise subpicture allocation for each atlas is conducted by the DWS allocator, as shown in Figure 1. The DWS allocation process has two main functions: (i) the separation of each BV and AV patch, and (ii) the decoder-wise subpicture merging into a single bitstream. The first

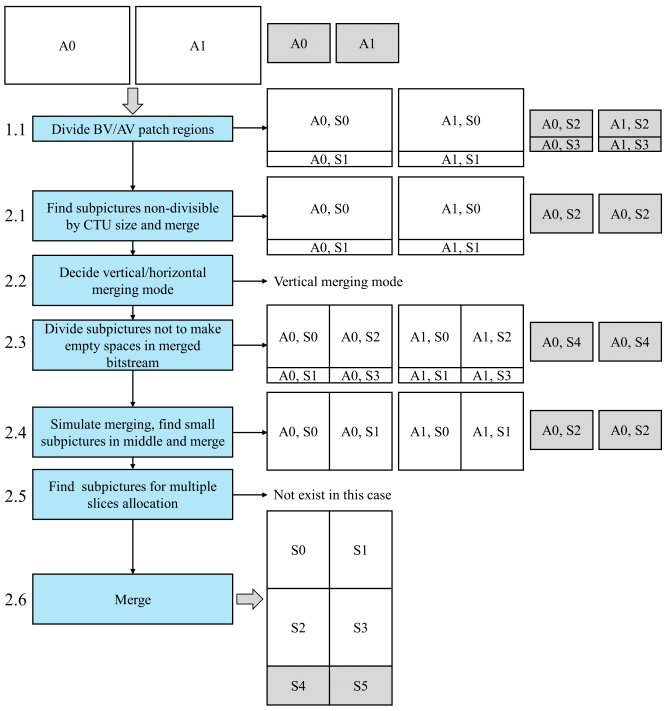

Fig. 3: Partitioning, extracting, and merging example of the proposed DWS-BEAM method Abbreviations: A, atlas; S, subpicture

function is to apply selective streaming (e.g. a user's viewport-aware streaming). Sending high-quality BVs which belong to the user's viewport can improve the user's quality of experience (QoE), while others are of low-quality, by dividing each BV into subpictures. Further, for selective rate-distortion optimisation (RDO) streaming, the separation of BV/AV patch areas is useful. According to a standard proposal in MPEG-I, asymmetric QP allocation on basic and AV patches showed Bjøntegaard Delta-rate (BD-rate) savings [14]. By using the tile/subpicture in HEVC/VVC, mixed-quality streaming can be performed for RDO without re-encoding and transcoding. In particular, lowering the quality of the AV patches increases the viewport quality at the same bitrate because BV patches carry most of the information; BV's quality is more important than AV's quality. As a result, the BV/AV patch regions are divided by the DWS allocator, as shown in stage 1.1 in Figure 3. The second function is to enable subpicture merging into a single bitstream, considering the following five conditions:

1) Subpicture width and height with respect to the coding tree unit (CTU) size,
2) Selection of vertical or horizontal merging mode,
3) Preventing empty spaces in the output picture,
4) Subpicture height restrictions when merging in VVC,
5) Relationship between tile/subpicture and slice in HEVC/VVC.

To meet the first condition, the DWS allocator finds subpictures whose width and height are not a multiple of the CTU size, typically 64×64 in HEVC and 128×128 in VVC.

HEVC and VVC do not allow tile/subpicture merging if the resolutions of the input pictures are not divisible by the CTU size. Therefore, if two adjacent subpictures are not divisible by the CTU size, the DWS allocator checks if the sum of the subpictures' width and height are divisible by the CTU size. If divisible, the DWS allocator removes the two subpictures and groups them into one subpicture, although BV patches and AV patches are located in the same subpicture. In stage 2.1 in Figure 3, S_2 and S_3 are merged because S_3 is not divisible by the CTU height. To meet the second condition, the DWS allocator decides whether the subpictures are merged horizontally or vertically by comparing the width and height of the atlas. If the width is larger than the height, the subpictures are vertically merged, and the geometry subpictures are located at the bottom of the texture subpictures. Otherwise, the horizontal merging is performed. Without considering the second condition, the resolution of the output picture can be horizontally or vertically too large, which is not compatible with low-end devices. In stage 2.2 in Figure 3, the texture atlas has a resolution of 4096×2176, and the width is larger than the height. Therefore, the subpictures are merged vertically. The third condition must be satisfied because the MIV was designed to minimise the pixel rate to use fewer decoders, thereby empty spaces in the merged picture, as shown in Figure 2a, are against the MIV. In stage 2.3 in Figure 3, subpictures from texture atlases are divided vertically to allocate the geometry subpictures at the bottom of the texture subpictures, as shown in stage 2.6. Otherwise, one geometry subpicture must be located under another geometry subpicture, which generates empty spaces. The fourth condition is for VVC subpicture merging, whereas there are no such restrictions for HEVC. If subpictures at the middle of the merged picture have heights which are smaller than those of the subpictures at the bottom, VVC does not allow merging. Therefore, the DWS allocator simulates the merging after satisfying the third condition. If a subpicture in the middle has a smaller height than the bottom subpictures, the DWS allocator groups the middle and upper subpictures into one subpicture. For example, in stage 2.3 from Figure 3, S_1, S_3 are located in the middle row during the merging. Because S_4 is bigger than the upper subpictures, the VVC subpicture merger cannot merge. Locating S_0, S_3 to the bottom does not work in some cases when S_2 is not divisible by the CTU size. Therefore, the DWS allocator groups S_0, S_1 into one subpicture, and also for S_2, S_3. The fifth condition is important for compatibility with the existing devices. Typically, one slice exists in the tile/subpicture; multiple slices exist in one tile/subpicture only if they are vertically aligned [15]. However, the use case shown in Figure 3 does not have subpictures in which multiple slices can be located, by so stage 2.5 is skipped in this case. Finally, at stage 2.6, all subpictures are merged into a single bitstream that is compatible with VVC. Still, the subpicture allocation by the proposed DWS-BEAM can transmit four bitstreams like the current TMIV does, and the proposed method can merge bitstreams considering the client's computational availability without re-encoding and transcoding.

IV. Experimental Results

A. Experimental Conditions

This section describes the experimental conditions. This experiment was conducted under common test conditions (CTCs) for MIV [16]. TMIV version 8.0 [17] was used as described in the CTC document. VVC test model (VTM) version 11.0 [18] was used for encoding, extracting, and merging, as specified in the exploration experiments (EEs) on future MIV [19]. Meanwhile, the VTM 11.0 subpicture merger cannot merge subpictures whose height is not divisible by the CTU height. Therefore, a simple modification, as shown in Equation 1, was applied:

$$pic_{h_{CTU}} = \lfloor (pic_h + CTU_h - 1)/CTU_h \rfloor \tag{1}$$

where $pic_{h_{CTU}}$ is the height of the merged picture in the CTU unit, pic_h is the height of the merged picture in the luma samples, and CTU_h is the height of the CTU in luma samples. For the decoding time measurement, Fraunhofer versatile video decoder (VVdeC) version 1.1.1 [20] was used to simulate real-time decoding. To evaluate the performance of the methods, the immersive video PSNR (IV-PSNR) were used [21]. In this experiment, six immersive video test sequences were used. The test sequences, *ClassroomVideo*, *Museum*, *Painter*, *Chess*, *ChessPieces*, and *Hijack* were classified according to their class names, computer-generated (CG)-A, CG-B, natural content (NC)-D, CG-N, CG-Q, and CG-C, respectively. The immersive video server used for this experiment had two Intel Xeon E5-2687w v4 CPUs (24 cores and 48 threads), 128 GB of memory, and Ubuntu 18.04 installed. The edge server had an Intel i7-7700k processor (4 cores and 8 threads), 16GB of memory, and Windows 10 installed. For the client, an edge server with the same machine specifications was used. For the control group, the VTM anchor, which encodes the atlases according to CTC, was used. For the experimental group, m54274, m56827, and the proposed DWS-BEAM were used. To test m56827, MCTS and subpicture BEAM were used to assign different quantisation parameters (QPs) for texture and geometry. Note that the BEAM was used because m56827 merges texture and geometry in the pixel domain, and the current VTM and versatile video encoder (VVenC) do not support subpicture level asymmetric QP allocation in a picture, therefore each atlas was encoded using subpictures, extracted, and merged following the specifications of m56827. The combination of m56827 and BEAM was defined as m56827 + BEAM in this experiment. In the subpicture encoding, one tile was assigned to one subpicture. Meanwhile, the VVC subpicture merger recommends disabling joint chroma coding, ALF, CCALF, LMCS, and AMaxBT options for subpicture merging. Thus, a VTM anchor to disable the aforementioned five options was added to the experimental group and defined as a VTM anchor (JACLA off) in this experiment. Furthermore, the experiments using subpictures disabled the JACLA options for the same reason.

TABLE I: Decoding runtime ratio compared to the VTM anchor

Class	VTM anchor	VTM anchor (JACLA off)	m54274 [11]	m56827 [13]+BEAM	DWS-BEAM (proposed)
CG-A	100.0%	99.3%	90.4%	**78.1%**	97.1%
CG-B	100.0%	101.0%	104.4%	85.8%	**78.9%**
NC-D	100.0%	100.9%	105.5%	88.2%	**76.7%**
CG-N	100.0%	100.6%	107.2%	95.8%	**82.9%**
CG-Q	100.0%	100.5%	106.0%	88.0%	**71.3%**
CG-C	100.0%	98.8%	93.9%	**83.1%**	97.8%
Average	100.00%	100.04%	100.92%	86.25%	**84.22%**

TABLE II: IV-PSNR BD-rate performances compared to the VTM anchor (Negative value indicates bitrate saving compared to the anchor)

Class	VTM anchor	VTM anchor (JACLA off)	m54274 [11]	m56827 [13]+BEAM	DWS-BEAM (proposed)
CG-A	0.00%	−0.29%	−1.59%	−1.60%	**−2.33%**
CG-B	0.00%	−1.61%	−2.63%	**−3.03%**	−2.08%
NC-D	0.00%	**−2.30%**	−1.82%	−2.09%	−2.27%
CG-N	0.00%	−0.28%	−0.84%	**−1.00%**	0.13%
CG-Q	0.00%	−0.47%	−1.33%	**−2.61%**	−0.74%
CG-C	0.00%	**−2.43%**	−1.07%	−1.01%	−0.92%
Average	0.00%	−1.28%	−1.46%	**−1.87%**	−1.23%

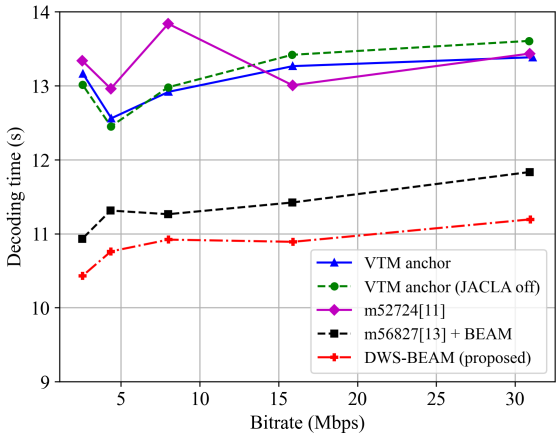

Fig. 4: Decoding runtime

TABLE III: Pixel rate, number of supported decoder instances

Items	VTM anchor (JACLA on/off)	m54274 [11]	m56827 [13]+BEAM	DWS-BEAM (proposed)
Pixel rate [GP/s]	**0.67**	0.77	**0.67**	**0.67**
No. of decoders	4	2, 4	2, 4	**1, 2, 4**

B. Analysis of the Results

This section explains and analsses the experimental results of these methods. Table I shows the decoding runtime ratio compared to the VTM anchor. Among the methods, the proposed DWS-BEAM showed the highest decoding runtime savings, as shown in Figure 4. Notably, the proposed method showed 23.3% gain in NC-D, which implies a higher efficiency on the natural content. Because the tile/subpicture can be decoded independently and parallel processing can be applied, decoding performance on the subpicture shows gain.

To evaluate the compression efficiency in terms of quality, the BD-rate [22] was used, where negative values represent the amount by which the bitrate has decreased for the same quality, whereas positive values indicate an increase in the bitrate. Table II shows the IV-PSNR BD-rate compared to the VTM anchor. Methods using subpictures show gains due to the activation of the MCTS. Because the MCTS turns off the in-loop filter options across the tiles/subpictures, there is a BD-rate gain compared to the anchor [23], thus BD-rate gain can occur when using subpictures in MIV. This is because each AV patch contains different shapes and colours from different views, and the deblocking filter causes a decrease in the quality of the details. Furthermore, IV-PSNR was designed to ignore the view synthesis artifacts that are not noticeable to humans, and the IV-PSNR BD-rate gain is advantageous for subjective quality. Table III lists the pixel rate and number of supported decoder instances. Because the MIV standard aims to reduce the number of pixels, the pixel rate is an important factor. All the methods except m54274 had the same pixel rate (0.67 GP/s) because there are empty areas in the merged bitstream generated by m54274. Further, the proposed DWS-BEAM is advantageous because it can provide one, two, or four bitstreams, depending on the client's specifications without re-encoding or transcoding. This allows a flexible streaming system using an edge server and requires less server-side storage. Therefore, the proposed DWS-BEAM method is advantageous for the practical and real-time implementation of MIV.

V. Conclusion

This paper proposed a decoder-wise subpicture bitstream extracting and merging (DWS-BEAM) method for MIV. Specifically, the atlases generated by the TMIV were divided into subpictures considering selective streaming and flexible merging for single-, dual-, and quad-pass decoding without re-encoding and transcoding. The proposed DWS-BEAM method showed 1.23% IV-PSNR BD-rate savings and 15.78% decoding runtime savings as compared to the VTM anchor. The proposed method is simple, efficient, and compatible with the existing HEVC and VVC standards. In the future work, extensive experiments for optimal QP allocation for each subpicture will be conducted.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00231-002, Development of Low Latency VR·AR Streaming Technology based on 5G edge cloud). This research was also supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2021-2017-0-01630) supervised by the IITP(Institute for Information & communications Technology Promotion).

REFERENCES

[1] 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.

[2] D. V. Nguyen, T. T. Le, S. Lee, and E.-S. Ryu, "Shvc tile-based 360-degree video streaming for mobile VR: PC offloading over mmWave," *Sensors*, vol. 18, no. 11, p. 3728, 2018.

[3] J.-B. Jeong, S. Lee, I.-W. Ryu, T. T. Le, and E.-S. Ryu, "Towards Viewport-dependent 6dof 360 Video Tiled Streaming for Virtual Reality Systems," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3687–3695.

[4] J.-B. Jeong, S. Lee, I. Kim, S. Lee, and E.-S. Ryu, "Implementing VVC Tile Extractor for 360-degree Video Streaming Using Motion-Constrained Tile Set," *Journal of Broadcast Engineering*, vol. 25, no. 7, pp. 1073–1080, 2020.

[5] T. Thanh Le, J.-B. Jeong, S. Lee, J. Kim, and E.-S. Ryu, "An Efficient Viewport-Dependent 360 VR System Based on Adaptive Tiled Streaming," 2021.

[6] J. M. Boyce, R. Doré, A. Dziembowski, J. Fleureau, J. Jung, B. Kroon, B. Salahieh, V. K. M. Vadakital, and L. Yu, "MPEG Immersive Video Coding Standard," *Proceedings of the IEEE*, 2021.

[7] G. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[8] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, "Developments in international video coding standardization after avc, with an overview of versatile video coding (vvc)," *Proceedings of the IEEE*, 2021.

[9] "Thoughts on Immersive Media Decoding Interface for VVC." Standard ISO/IEC JTC1/SC29/WG11, MPEG/n18438, 2019.

[10] V. K. M. Vadakital, "Description of Immersive Video Core Experiments 1: Bitstream Adaptation." Standard ISO/IEC JTC1/SC29/WG11, MPEG/n19215, 2020.

[11] L. Kondrad, V. K. M. Vadakital, and L. Ilola, "CE-1.3: frame packed video sub-bitstream type." Standard ISO/IEC JTC1/SC29/WG11, MPEG/m54274, 2020.

[12] A. Hallapuro and M. M. Hannuksela, "AHG3/AHG12: Subpicture merging software." ITU-T SG 16 WP3 ISO/IEC JTC1/SC29/WG11, JVET-S0162, 2020.

[13] B. Salahieh, G. Naf, and J. Boyce, "Frame Packing Implementation in TMIV." Standard ISO/IEC JTC1/SC29/WG4, MPEG/m56827, 2021.

[14] S. Lee, J.-B. Jeong, and E.-S. Ryu, "[MPEG-I Visual] Report on Asymmetric Quantization on MIV." Standard ISO/IEC JTC1/SC29/WG11, MPEG/m55014, 2020.

[15] A. Zare, M. Homayouni, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "6K and 8K Effective Resolution with 4K HEVC Decoding Capability for 360 Video Streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 2s, pp. 1–22, 2019.

[16] J. Jung and B. Kroon, "Common Test Conditions for MPEG Immersive Video." Standard ISO/IEC JTC1/SC29/WG4, MPEG/n0051, 2020.

[17] Test model for immersive video (TMIV) v8.0. [Online]. Available: https://gitlab.com/mpeg-i-visual/tmiv/-/tree/v8.0

[18] VVC test model (VTM) 11.0. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-11.0

[19] J. Jung, V. K. M. Vadakital, and D. Mieloch, "Exploration Experiments on Future MPEG Immersive Video." Standard ISO/IEC JTC1/SC29/WG4, MPEG/n0055, 2021.

[20] Versatile video decoder (VVdeC) v1.1.1. [Online]. Available: https://github.com/fraunhoferhhi/vvdec/tree/v1.1.1

[21] A. Dziembowski and M. Domański, "[MPEG-I Visual] Objective quality metric for immersive video." Standard ISO/IEC JTC1/SC29/WG11, MPEG2019/m48093, 2019.

[22] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, 2001.

[23] E. Yeo, S. hyo Park, and J.-W. Kang, "[MPEG-I Visual] Deblocking Filter for Atlases in Basic and Additional Views." Standard ISO/IEC JTC1/SC29/WG11, MPEG/m55227, 2020.