VVC Subpicture-Based Frame Packing for MPEG Immersive Video

JONG-BEOM JEONG, (Graduate Student Member, IEEE), SOONBIN LEE, (Graduate Student Member, IEEE), AND EUN-SEOK RYU, (Senior Member, IEEE)

Department of Computer Science Education, Sungkyunkwan University (SKKU), Seoul 03063, South Korea

Corresponding author: Eun-Seok Ryu (esryu@skku.edu)

This work was supported in part by the Institute of Information and communications Technology Planning and Evaluation (IITP) Grant through the Korea Government [Ministry of Science and Information and Communications Technology (MSIT)], Foreground and Background Matching 3D Object Streaming Technology Development, 50%, under Grant 2022-0-00981; and in part by IITP Grant through the Korea Government (MSIT), Development of Ultra High Resolution Unstructured Plenoptic Video Storage/Compression/Streaming Technology for Medium to Large Space, 50%, under Grant 2020-0-00920.

ABSTRACT The moving picture experts group (MPEG) video coding group started an immersive video coding standard project to accomplish acceptable bandwidth and decoding resources for multiple immersive videos with texture (color) and geometry (depth) information. The MPEG immersive video (MIV) coding standard provides functionality to remove inter-view redundancy and merge the residuals into videos defined as atlases. Although MIV reduces the number of videos to decode, an efficient decoder instance reduction method has not been actively investigated. This paper proposes a frame packing implementation in MIV software. The proposed method conducts frame packing for geometry atlases to align with texture atlases, encodes them as subpicture bitstreams, and merges one texture/geometry atlas into a single bitstream with a bitstream merger. Furthermore, the proposed method provides control over the number of decoder instances without significantly increasing the computational complexity compared to existing frame packing methods. Therefore, the proposed method was accepted in MIV and will be implemented on the recent version of the reference software of MIV.

INDEX TERMS Virtual reality, metaverse, six degrees of freedom, MPEG immersive video, versatile video coding, subpicture, tile, frame packing.

I. INTRODUCTION

With the growing demand and interest in augmented reality (AR) and virtual reality (VR) streaming technologies including acquisition, pre-processing, encoding, decoding, post-processing, and rendering have become important. Compared with traditional 2-D video streaming systems, an immersive video streaming system for a head-mounted display (HMD) requires a higher video quality and computational complexity. For instance, [1] mentioned that to minimize the motion sickness and increase immersion, a video which provides 90 frames per second (FPS), a resolution of 12K, and 20ms for motion-to-photon (MTP) latency needs to be streamed. The moving picture experts group (MPEG), which deals with immersive media standardization, established the MPEG-immersive (MPEG-I) subgroup to define use cases and metadata for immersive videos. Suppose a user is sitting on a chair and watching a 360-degree video through an HMD captured with a single camera. The HMD can reflect the user’s head rotation and render videos based on the rotation. However, this use case cannot render videos that reflect the user’s body movement. Therefore, this approach was defined as three degrees of freedom (3DoF) because it only provides freedom towards yaw, pitch, and roll. For 3DoF video streaming, the viewport-dependent streaming approaches has been actively discussed to save bandwidth and latency [2], [3]. In high-efficiency video coding (HEVC), a tile representing a rectangular area in a picture facilitated to facilitate parallel processing [4]. Tiles in a picture can be independently encoded, extracted, and decoded using a motion-constrained tile set (MCTS) [5]. The viewport-dependent 360-degree video streaming method exploits MCTS to extract and
transmit high quality tiles that represent the field of view (FoV) and low quality videos that covers omnidirectional field, saves bandwidth and latency.

To provide motion parallax to the user with an HMD is to capture natural or computer-generated (CG) contents using multiple cameras, transmit them, and synthesize virtual views corresponding to the user’s movements. MPEG-I defines this use case as six degrees of freedom (6DoF), which supports movements towards the yaw, pitch, roll, x, y, and z axes. MPEG immersive video (MIV) coding standard [6] presents immersive video compression, transmission, and synthesis methods, accomplishing better quality and fewer decoder instances than the existing multi-view coding standards such as multi-view HEVC (MV-HEVC) [7]. MIV removes the inter-view correlation of immersive video using depth information from the geometry videos and color differences from the texture videos. The MIV then extracts the residuals and merges them to reduce the number of pixels (pixel rate). Consequently, it generates output videos defined as atlases. Recent research on MIV has focused on efficient compression methods for immersive videos, accomplishing quality improvement over the existing MIV [8], [9], [10], [11], [12], [13], [14]. Although MIV recommends limiting the number of decoder instances to four, four decoder instances is a challenge for legacy systems that normally contains one hardware decoder. [15] and [16] proposed frame packing methods for MIV to reduce the decoder instances by merging one texture and one geometry atlases. Despite their effectiveness in reducing the decoder instances, these approaches are inappropriate for application to MIV because the current video compression reference software does not support the use cases.

This paper proposes a frame packing method for a decoder-adaptive 6DoF immersive video streaming system. Fig. 1 shows the proposed frame packing method using MIV, with the proposed components shown in blue. Immersive videos, each of which contains texture and depth with camera parameters are inputted to the MIV encoder. In the MIV encoder, the inter-view redundancy of immersive videos is removed by a pruner, and the residuals are extracted and packed into atlases by a patch packer. The proposed geometry packer divides geometry atlases vertically and packs them; therefore, packed geometry and texture atlases have same heights. The video encoder encodes the atlases using MCTS and generates video bitstreams, which are sent to the edge server with metadata generated by the MIV encoder. The MIV metadata follows the visual volumetric video coding (V3C) standard, which is also compatible with the video-based point cloud coding (V-PCC). Depending on the specifications of the video decoder at the client, for instance, if the client is a legacy device, the bitstream merger at the edge server merges the bitstreams; therefore, a merged bitstream contains texture and depth. The client receives the bitstreams and the video decoder decodes them. The decoded pictures and metadata are provided to the MIV decoder, and the geometry unpacker unpacks the packed atlases and sends them to the renderer. The renderer synthesizes a virtual view corresponding to the user’s viewport, and finally the synthesized videos are displayed on the HMD.

The remainder of this paper is organized as follows. Section 2 presents the background focusing on MIV, and Section 3 explains related work. Section 4 describes the proposed method, and Section 5 presents the experimental conditions, results, and analysis. Finally, Section 6 provides the conclusion and presents insights towards decoder-adaptive 6DoF immersive video streaming.

II. BACKGROUND
Multiple video compression is not a new research topic and has been continuously discussed [17], [18], [19]. Embedding
the inter prediction across multiple views into existing video compression standards enhances compression performance. For instance, MV-HEVC was established as an extension of HEVC, which exploits the existing coding methods of HEVC and applies multiview-specific coding methods. However, MV-HEVC has disadvantages for unstructured multiple videos and compatibility. Because MV-HEVC is a block-based coding standard, it can efficiently compress multiple videos obtained by 1-D or 2-D camera arrays without guaranteeing the same efficiency for unstructured multiple videos where rotations and positions of cameras differ. Further, multiview-specific coding methods were applied to codecs; therefore, the existing acceleration methods cannot be directly applied.

MPEG-I issued a call for proposals (CIP) in 2019 for three degrees of freedom plus (3DoF+), a limited version of 6DoF [20]. The CIP applies a warping-based inter-view redundancy removal method, which is efficient for unstructured immersive videos, uses existing 2-D video coding standards for compatibility, and reduces the pixel rate and number of decoder instances for low-end devices. The MPEG-I experts established the MIV and the test model for immersive video (TMIV) as the reference software based on the responses [21], [22], [23], [24], [25]. Fig. 2 shows the videos processed by the MIV encoder in the MIV mode. The MIV encoder supports two modes: MIV and MIV view modes. The MIV mode removes inter-view redundancy, and the MIV performance tests are based on this mode. The MIV view mode does not remove inter-view redundancy but selects representative views among the source views. In Fig. 2, the source views are the input immersive videos, distinguished by the basic and additional views. The inter-view redundancy is removed from the additional views, whereas the basic views preserve redundancy. The pruned views represent views after inter-view redundancy removal process, defined as pruning in MIV. Then, the residuals are extracted and merged into atlases. The inpainted background view is synthesized by the MIV encoder to efficiently represent the background, which is also included in the atlases. The common test conditions (CTC) of the MIV sets the number of atlases to four [26]: considering decoder-side constraints, implying that decoder-side aspects are extremely important in MIV. The latest CTC of MIV recommends using a versatile video encoder (VVenC) [27], which is an optimized software of the versatile video coding (VVC) standard [28].

III. RELATED WORK

Section 2 explained that decoder-side constraint is amongst the most important aspects of MIV. In MPEG-I, two standard proposals were submitted and frame packing methods were presented to reduce the number of decoder instances [15], [16]. Fig. 3 shows the frame packed atlases of [15] and [16] and their limitations. [15] proposed a subpicture-based frame packing method, as shown in Fig. 3a. Texture and geometry atlases were encoded as 1 × 1 subpictures using VVenC, and the subpicture bitstreams were then merged using the SubpicMergeApp of the VVC test model (VTM). A geometry bitstream was packed under a texture bitstream, as shown in Fig. 3a. The geometry atlases were downsampled by 2 × 2 to reduce the pixel rate. The frame packing method of [15] has two limitations. First, the subpicture bitstreams cannot be merged in Fig. 3a because the height of the texture is not a multiple of the coding tree unit (CTU) size. In VVC, all subpicture bitstreams must contain pictures that can be divided by the CTU size, for instance, 128 in VVC. Therefore, the subpicture bitstreams cannot be merged. Second, because a geometry bitstream is packed under a texture bitstream and their sizes are different, the merged bitstream contains empty areas that are not covered by subpicture bitstreams. This causes decoding errors, implying that the frame packing in [15] is not compatible with some existing standard softwares. [16] proposed a pixel-domain frame packing method, as shown in Fig. 3b. A geometry atlas is divided horizontally and packed under a texture atlas in a pixel domain, then the packed video is encoded using VVenC. The current CTC of the MIV recommends using different quantization parameters (QPs) for texture and geometry atlases because the geometry exhibits different characteristics from those of the texture. Therefore, VVenC must assign different QPs for the texture and geometry regions. There are three limitations to the frame packing method of [16]. First, VVenC does not provide a functionality to assign different QPs for different regions without subpictures or tiles. The proposed method assumed to be advantageous because it does not rely on the MCTS. In HEVC, there is an optimized encoder Kvazaar [29] that provides functionality to use different QPs at the CTU level. Accordingly, MCTS is not required if MIV uses Kvazaar. However, VVenC does not provide this function without MCTS. Second, even if subpictures or tiles are used, VVenC cannot assign different QPs to a picture. The VVC itself does not forbid different QP allocations using subpictures or tiles, but currently, there are no VVC implementations that

![FIGURE 2. Source views, pruned views, and atlases generated by the MIV encoder using the MIV mode.](image-url)
provides this functionality. Third, as explained in Fig. 3b, the subpicture height is not divisible by the CTU size.

IV. VVC SUBPICTURE-BASED FRAME PACKING FOR MPEG IMMERSIVE VIDEO

This section presents the proposed geometry packing method for MIV, compatible with existing standards and their implementations. As explained in Section III, the existing packing methods have the following limitations: i) the subpicture width or height should be divisible by the CTU size, ii) empty spaces should not exist in the merged picture, and iii) different QPs must be assigned to texture and geometry atlases, and different QP allocations are not supported by the current VVC implementations in the same picture. Therefore, the proposed packing method focuses on compatibility with existing coding standards, including MIV, HEVC, and VVC. Previous packing methods packed the geometry atlas under the texture atlas, causing subpicture merging errors because, in some cases, the texture height was not a multiple of the CTU size. If the geometry atlas is packed under the texture atlas, adding a padded area under the texture to satisfy the CTU size constraint can be applied, as presented in [30]. However, adding a padded area that increases the number of invalid pixels is undesirable because the MIV was designed to reduce the pixel rate. Otherwise, packing the geometry atlas to the right side of the texture atlas can be a solution to detour the padding issue. The current TMIV calculates the atlas sizes by considering the pixel rate constraints and minimum block size. In addition, the width of the atlas is same with that of the input source views. Given widely used video widths (e.g. 1280 × 720, 1920 × 1080, 3840 × 1920) are divisible by the CTU size (e.g. 128 in VVC), packing the geometry atlas to the right side of the texture atlas does not require padding. Therefore, the proposed method packs the geometry atlas to the right side of the texture atlas.

As mentioned in [16] and Section III, empty spaces in the merged picture increases the pixel rate and causes decoding errors in some video coding standard implementations. The proposed packing method divides the geometry atlases vertically as well as packs vertically. Fig. 4 describes the proposed geometry packing and bitstream merging, focusing on the V3C bitstream (MIV metadata) considering compatibility with the MIV and VVC. The size of the texture atlas is represented by *vps_frame_width* and *vps_frame_height* in the V3C parameter set. The downsampling geometry atlas size is calculated by referencing *asme_geometry_scale_factor_x_minus1*.
FIGURE 4. Frame packing and bitstream merging considering V3C bitstream (MIV metadata).

and asme\_geometry\_scale\_factor\_y\_minus1 in the atlas sequence parameter set. Number of regions in the geometry atlas is the same as the geometry scaling factor on the y axis. For instance, if the geometry atlas is downsampled by two on the y axis, the number of regions is 2. Similarly, the width of each regions of the geometry atlas can be calculated using Eq. 1:

\[
W_{\text{region}} = \frac{W_{\text{atlas}} \times \text{GSF}_x}{N_{\text{region}}} + 1
\]

where \(W_{\text{region}}\), \(W_{\text{atlas}}\), \(\text{GSF}_x\), and \(N_{\text{region}}\) represents the width of the geometry atlas region, the width of the atlas, geometry scaling factor in the x axis, and the number of regions, respectively. Mentioned variables are stored as \(\text{pin\_region\_width\_minus1}\), \(\text{vps\_frame\_width}\), \(\text{asme\_geometry\_scale\_factor\_x\_minus1}\), and \(\text{pin\_regions\_count\_minus1}\) in the V3C bitstream, respectively. In Fig. 4, the width of each geometry atlas regions is 512, and the number of regions in the geometry atlas is two. Following these values, the geometry atlas was divided and packed vertically. Texture atlases and packed geometry atlases were encoded using VVENC as 1×1 subpictures. Different QPs can be applied because they can be encoded individually. The bitstream merger in Fig. 4 performs two functions: i) merges the subpicture bitstreams, and ii) modifies the V3C bitstream corresponding to the merged bitstream. For instance, \(\text{pin\_region\_top\_left\_x}\) and \(\text{pin\_region\_top\_left\_y}\), which indicate the top left x and y coordinates of each region, respectively, need to be modified. Furthermore, in the current VVC subpicture merger [31], all subpictures should be divisible by the CTU size. However, in HEVC, tiles or subpictures in the last column or row do not need to be divisible by the CTU size. This paper experimented on the last column and row subpictures regarding CTU size, and it was proved that subpictures in the last column or row don’t need to be divisible by the CTU size in the VVC. Therefore, Eq. 2 must be applied to the VVC subpicture merger.

\[
H_{\text{subpicture}} = H_{\text{CTU}} \cdot \left\lfloor \frac{H_{\text{subpicture}} + H_{\text{CTU}} - 1}{H_{\text{CTU}}} \right\rfloor
\]

where \(H_{\text{subpicture}}\) and \(H_{\text{CTU}}\) represents the height of each subpictures and the height of the CTU, respectively. This modification is for the subpicture merger; therefore, the proposed method remains compatible with the existing VVC standard.

Therefore, the proposed geometry packing and bitstream merging methods are compatible with MIV and VVC, providing functionality to reduce the number of decoder instances. Table 1 summarizes the comparisons of frame packing methods. The packing domain indicates whether the uncompressed atlases are packed in a pixel-domain or whether the atlases are encoded and then packed in a bitstream-domain using the subpicture merger. [15] used bitstream-domain frame packing, [16] applied a pixel-domain approach. There are two types of the proposed methods: i) geometry packing without bitstream merging (GP), which conducts pixel-domain frame packing only and ii) geometry packing and bitstream merging (GPM), which processes pixel-domain and bitstream-domain frame packing. The frame packing method in [15] increased the pixel rate of the merged pictures by 20%; therefore, some legacy devices that supports a low profile may be unable to decode the packed
TABLE 1. Comparisons of the frame packing methods. Abbreviations: GP, geometry packing without bitstream merging; GPM, geometry packing and bitstream merging.

| Feature                  | [15]          | [16]          | Proposed GP | Proposed GPM |
|--------------------------|---------------|---------------|-------------|--------------|
| Packing domain           | Bitstream-domain | Pixel-domain | Pixel-domain | Pixel + bitstream-domain |
| Pixel rate increase      | 20%           | 0%            | 0%          | 0%           |
| No. of encoders          | 4             | 2             | 4           | 4            |
| No. of decoders          | 2, 4          | 2             | 4           | 2            |
| VVenC encoding           | ○             | ×             | ○           | ○            |
| VVdeC decoding           | ×             | ×             | ○           | ○            |

FIGURE 5. Server-side storage, (a) frame packing method in [16], (b) the proposed geometry packing method. Abbreviations: A17, MIV main 17 frames; P17, frame packing 17 frames; GP17, geometry packing 17 frames; GPM17, geometry packing and bitstream merging 17 frames.

bitstreams. [16], and the proposed methods did not report any increase in pixel rate. Because [16] generates uncompressed packed atlases, two encoders and a corresponding number of decoders are required. [15] and the proposed methods required four encoders, but they could control the number of decoders by conducting subpicture merging optionally depending on the computational complexity of the client-side devices. The proposed GP and GPM methods require four and two decoders, respectively, where GP and GPM share same bitstreams and GPM conducts bitstream merging which requires a trivial increase in computational complexity. A recent version of VVenC (v1.4.0) [32] supports [15] and the proposed methods. However, packed atlases using [16] cannot be encoded using VVenC. This is because different QPs must be assigned within a packed atlas, and the current VVenC does not support this functionality. The merged bitstreams containing packed atlases obtained by the proposed method can be decoded by VVdeC.

The proposed method is also advantageous for server-side storage. Fig. 5 shows the server-side storage in the previous frame packing method and the proposed geometry packing method. The frame packing method in [19] was accepted and adopted in the current TMIV, denoted as P17. In the MIV main profile, denoted as A17, four atlases (two texture + geometry atlases) are generated. The previous frame packing method packs one texture and one geometry atlas into a packed atlas in the pixel domain; therefore, the merged bitstreams containing packed atlases obtained by the proposed method can be decoded by VVdeC.

The proposed method is also advantageous for server-side storage. The proposed method considers the CTU size and conducts packing to enable subpicture merging; therefore, the merged bitstreams containing packed atlases obtained by the proposed method can be decoded by VVdeC.

VOLUME 10, 2022
bitstreams. This is because the proposed method is based on subpictures, and bitstream extraction and merging are supported, thereby requiring a trivial increase in computational complexity. From the immersive video provider’s perspective, server-side storage is a substantially important aspect, and the proposed method is advantageous for server-side storage and compatibility with the existing coding standards.

V. EXPERIMENTAL RESULTS

This section explains the experimental conditions, results and subsequent analysis. MPEG-I defined the CTC of the MIV for fair evaluations, and this experiment also followed the recent version of CTC. Because the previously proposed frame packing methods [15], [16] encountered difficulties in conducting experiments on existing coding standards, the experiment was conducted on the MIV anchor and the proposed packing method.

A. EXPERIMENTAL CONDITIONS

This section explains the experimental conditions based on the CTC of MIV [26], which was published in 2022. Table 2 shows the software used in this experiment, which is listed in CTC. For MIV encoding and decoding, TMIV version v11.0 was used, and the proposed geometry packing was implemented on top of it. For video encoding and decoding, VVenC version v1.4.0, and VVdeC version v1.0.1, were used [33]. As the VVenC version v0.1.3.0, which is listed in the CTC, does not support subpictures, this experiment used the recent version of VVenC. The immersive video peak signal-to-noise ratio (IV-PSNR) is an end-to-end quality metric suitable for 6DoF immersive video quality assessment and which was used in this experiment [34]. A modified version of SubpicMergeApp in VTM version v11.0 conducted subpicture merging, and the modifications are explained in Section IV.

TABLE 2. Softwares used in the experiment.

| Software | Version |
|----------|---------|
| TMIV     | v11.0   |
| VVenC    | v1.4.0  |
| VVdeC    | v1.0.1  |
| IV-PSNR  | v3.0    |
| VTM      | v11.0   |

Table 3 lists the test conditions. A17 presents the experimental results for the MIV mode on 17 frames, which were set to the anchor. Because CTC recommends conducting experiments on 17 frames to investigate the coding efficiency for a group of pictures (GOP), which is typically 16 frames + 1 frame, this experiment also followed the recommendation. The required number of decoder instances is four because the MIV mode generates four atlases. Detailed descriptions of the MIV mode are provided in Section II. GP17 implies that geometry packing was applied to the MIV mode for 17 frames; therefore, the number of decoder instances did not change. GPM17 performs the subpicture merging on GP17; therefore, the required number of the decoder instances were two. This experiment compared GP17 and GPM17 to investigate the reduced bitrate caused by subpicture merging and video decoding/MIV decoding runtimes for different numbers of decoder instances.

The SubpicMergeApp recommends five disabled coding options: joint chroma coding, adaptive loop filter (ALF), cross component ALF (CCALF), luma mapping with chroma scaling (LMCS), and AMaxBT. These five options were disabled for A17, GP17 and GPM17. Because the deactivation of the above-mentioned five options decreases coding efficiency, experiments on the intra block copy (IBC) mode were conducted to increase coding gain. The IBC computes the motion estimation and compensation in the intra coding mode, which is efficient when several similar blocks exist in a picture [35]. The atlases contain similar objects from different source views and using IBC can increase the coding efficiency. In the HEVC coding standard, IBC was included in the screen content coding (SCC) extension to increase the coding gain for computer-generated content. In VVC, IBC can be used without an extension standard.

An Ubuntu server was used, which had one AMD EPYC 7713 64-Core Processor (64 cores and 128 threads) and 512 GB of memory. The edge server had an Intel i7-7700k processor (four cores and eight threads), 16GB of memory, and Windows 10. An edge server with the same machine specifications was used as the client.

For a fair evaluation, this experiment used nine mandatory immersive video test sequences listed in CTC [26]. The sequences are classified according to their class names, where CG stands for computer-generated content and NC represents natural content. Because of the difficulty in capturing immersive videos in natural environment, seven out of nine test sequences were captured in CG.

To measure the Bjøntegaard delta rate (BD-rate) gain for quality, five bitrate points and the corresponding QPs were determined, as explained in the CTC. Table 5 presents the texture QPs for each test sequences. The geometry QPs were calculated using Eq. 3.

\[ QP_g = \max(1, [-14.2 + 0.8 \cdot QP_t]) \] (3)

where \( QP_g \) represents QP for geometry and \( QP_t \) is QP for texture. Using Eq. (3), lower QPs were assigned to the geometry than to those of the texture to preserve depth values, which are important for virtual view synthesis. After TMIV encoding, video encoding, bitstream merging, and video decoding,
FIGURE 6. Packed atlases for sequence ClassroomVideo, Museum and Painter. The first and third columns represent atlases index 0, and the second and fourth columns represent atlases index 1.

TABLE 4. Specifications of the MIV test sequences.

| Name       | Class | Projection | Resolution (Source view) (Atlas) | No. of source views |
|------------|-------|------------|----------------------------------|--------------------|
| ClassroomVideo | CG-A  | Full-ERP   | 4096×2048 4096×2176              | 15                 |
| Museum     | CG-B  | Half-ERP   | 2048×2048 2048×2048 2048×2048 2048×2048 | 24                 |
| Fan        | CG-O  | Perspective | 1920×1080 1920×4640              | 15                 |
| Kitchen    | CG-J  | Perspective | 1920×1080 1920×4640              | 25                 |
| Painter    | NC-D  | Perspective | 2048×1088 2048×4352              | 16                 |
| Frog       | NC-E  | Perspective | 1920×1080 1920×4640              | 13                 |
| Carpark    | NC-P  | Perspective | 1920×1088 1920×4640              | 9                  |
| Chess      | CG-N  | Half-ERP   | 2048×2048 2048×2048 2048×2048 2048×2048 | 10                 |
| Group      | CG-R  | Perspective | 1920×1080 1920×4640              | 21                 |

TABLE 5. Texture QPs for the experiment. Target bitrate: 50Mbps (QP1), 28Mbps (QP2), 16Mbps (QP3), 9Mbps (QP4), and 5Mbps (QP5).

| Name       | QP1 | QP2 | QP3 | QP4 | QP5 |
|------------|-----|-----|-----|-----|-----|
| ClassroomVideo | 26  | 28  | 33  | 41  | 51  |
| Museum     | 29  | 38  | 44  | 48  | 51  |
| Fan        | 30  | 35  | 41  | 46  | 51  |
| Kitchen    | 17  | 24  | 29  | 35  | 41  |
| Painter    | 21  | 29  | 38  | 45  | 51  |
| Frog       | 28  | 32  | 37  | 42  | 46  |
| Carpark    | 22  | 26  | 33  | 40  | 47  |
| Chess      | 17  | 25  | 31  | 37  | 45  |
| Group      | 22  | 28  | 34  | 39  | 46  |

virtual views at each source view position were synthesized in the TMIV decoding stage. As shown in Fig. 5, GP17 indicates that only geometry packing was applied, and GPM17 indicates that both geometry packing and bitstream merging were applied. The decoder-side efficiency was measured by comparing the GP17 and GPM17.

B. EXPERIMENTAL RESULTS

This section presents the experimental results and provides insights into the results. Fig. 6 shows the packed atlases using the proposed geometry packing and bitstream merging methods, which were implemented on top of the TMIV software.
The figures in the first and third columns represent atlases of index 0, which consist mainly of basic views. The figures in the second and fourth columns represent atlases of index 1, which is composed of additional view patches. A17, which represents the MIV main profile, contains four atlases. The proposed GPM17, which conducts both geometry packing and bitstream merging, contains two packed atlases. Therefore, the frame packed geometry atlases are encoded and merged with the texture atlases.

Table 6 shows BD-rates on Y-PSNR and IV-PSNR of GP17 and GPM17 compared to A17. A high BD-rate was calculated using QP1 to QP4, while a low BD-rate used QP2 to QP5. Overall, the proposed method with geometry packing only showed BD-rate loss compared to the MIV main profile. Because of the MCTS and continuous information loss at the edge of the packed geometry regions, coding loss was observed compared to A17. However, as mentioned in [36], BD-rate losses from 3.18% to 5.73% were observed when enabling the MCTS on 360-degree video tiled streaming. The geometry packing and bitstream merging were enabled in GPM17, it showed 0.64% and 0.67% BD-rate improvements on low Y-PSNR and IV-PSNR, respectively. During the merging, the number of non-video coding layer (non-VCL) network abstraction layer (NAL) units such as video parameter set (VPS), sequence parameter set (SPS), and picture parameter set (SPP) were reduced; therefore, decreasing the number of bits to be transmitted.

Table 7 presents the MIV encoding, video encoding, video decoding, and MIV decoding runtime ratios of GP17 and GPM17 compared with those of A17. The client-side device uses a single decoder and this experiment describes the aggregated runtime results. GP17 and GPM17 share MIV encoding and video encoding processes; therefore, the runtime results are same. When geometry packing was applied, there was a 3.89% of runtime saving for MIV encoding. Thus, proving that geometry packing does not increase the MIV encoding runtime. For video encoding, a 11.36% runtime saving was observed, which is advantageous for real-time 6DoF immersive video streaming. GP17, which applied geometry packing only, showed a 14.94% decoding time saving compared with A17. This is because when utilizing
TABLE 8. BD-rates of GP17 + IBC and GPM17 + IBC on Y-PSNR and IV-PSNR compared to A17 as a MIV mode anchor. GP17 as the proposed method w/o bitstream merging and GPM17 as the proposed method with bitstream merging.

| Name      | GP17 + IBC | GPM17 + IBC |
|-----------|------------|-------------|
|           | High Y-PSNR | Low Y-PSNR | High IV-PSNR | Low IV-PSNR | High Y-PSNR | Low Y-PSNR | High IV-PSNR | Low IV-PSNR |
| Classroom | -0.12%      | -0.31%      | -0.46%       | -0.33%      | -1.11%      | -2.09%      | -1.53%       | -2.37%       |
| Museum    | -0.31%      | 0.02%       | -0.38%       | -0.05%      | -1.51%      | -1.95%      | -1.54%       | -1.98%       |
| Fan       | 0.10%       | 0.27%       | -0.43%       | -0.21%      | -0.20%      | -0.16%      | -0.73%       | -0.66%       |
| Kitchen   | 1.36%       | 0.70%       | -0.47%       | -0.22%      | 0.81%       | -0.19%      | -1.00%       | -1.11%       |
| Painter   | -0.06%      | 0.11%       | -0.19%       | -0.16%      | -0.50%      | -0.66%      | 0.32%        | -0.39%       |
| Frog      | 0.15%       | 0.10%       | 0.02%        | -0.10%      | -0.03%      | 0.49%       | -0.14%       | 0.49%        |
| Carpark   | -0.38%      | -0.31%      | -0.54%       | -0.35%      | -0.90%      | -1.18%      | -1.06%       | -1.23%       |
| Chess     | 0.39%       | 0.02%       | 0.08%        | 0.15%       | -0.44%      | -1.47%      | -0.73%       | -1.22%       |
| Group     | 0.20%       | -0.04%      | 0.31%        | 0.22%       | -0.15%      | -0.72%      | -0.03%       | -0.41%       |
| Average   | 0.18%       | 0.13%       | -0.23%       | -0.12%      | -0.45%      | -0.88%      | -0.72%       | -0.99%       |

Table 8 describes the BD-rates on Y-PSNR and IV-PSNR of GP17 + IBC and GPM17 + IBC compared with A17. The BD-rate improvements for Y-PSNR and IV-PSNR were 0.46% and 0.42%, respectively, compared to the results wherein the IBC was disabled. According to [38], the Y-PSNR and IV-PSNR BD-rate gain rates were 9.66% and 9.62%, respectively, when using IBC in HEVC-SCC. However, the experimental results of the IBC in VVC showed a BD-rate gain of below 1%. This is due to the constrained search range of IBC in VVC. Fig. 7 shows the reference areas for the IBC. A square with a hatch denotes current CTU, and gray squares represent candidate reference blocks for IBC. (a) reference areas in HEVC-SCC, (b) reference areas in VVC.

FIGURE 7. Reference areas for IBC. Each square represents a CTU.
reference area. Therefore, the IBC in VVC for the MIV does not provide high coding gains, but it is advantageous because the increased computational complexity due to the IBC is not high. Table 9 shows the MIV encoding, video encoding, video decoding, and MIV decoding runtime ratios of GP17 + IBC and GPM17 + IBC compared with A17. Compared to A17, only 4.07% and 0.18% increase of video encoding and MIV decoding runtime, respectively, were observed, when the IBC was enabled. Therefore, in the VVC, the IBC increases the BD-rate of the MIV with a small increase in the encoding runtime. Despite the coding loss when subpictures are used in VVC, using IBC can improve BD-rate performance.

VI. CONCLUSION

This paper proposed geometry packing and bitstream merging methods for 6DoF immersive video streaming based on MIV and VVC. The proposed method divides and packs geometry atlases. After geometry packing, each atlas was encoded as a subpicture using different QPs. Subsequently, one texture and one geometry atlas bitstreams were merged, and the packed geometry was packed to the right of the texture so as not to add padding. The proposed bitstream merging is lightweight and versatile, and this can be applied to both the server and client, with a trivial increase in computational complexity. Furthermore, the proposed method demonstrated 0.67% of the IV-PSNR BD-rate gain and 11.36% decoding runtime saving compared with the MIV main profile while providing functionality to control the number of decoder instances without increasing server-side storage. Furthermore, the experimental results with IBC showed an IV-PSNR BD-rate gain of 0.99% with a 4.07% increase in video encoding runtime. The proposed method was presented in MIV meeting and accepted; therefore, this will be implemented in the recent version of TMIV. In future work, an enhanced frame packing method using filler bitstream [39] will be researched and extensive experiments for merging atlases into a single bitstream and the corresponding metadata development will be conducted.

ACKNOWLEDGMENT

The authors would like to thank Bart Kroon from Philips, Basel Salahieh, Jill Boyce from Vimmerse, Lukasz Kondrad from Nokia, and Vinod Kumar Malamal Vadakital from Ofinno, who contributed to the frame packing methods in the MIV and provided useful comments. They would also like to thank Hyun-Ho Kim and Yong-Hwan Kim from KETI, who conducted crosscheck for their proposed method, and Gwangsoon Lee from ETRI, who provided us with constructive comments.

REFERENCES

[1] M.-L. Champel, T. Stockhammer, T. Fautier, E. Thomas, and R. Koenen, Quality Requirements for VR, document ISO/IEC JTC1/SC29/WG11, 116th MPEG meeting, MPEG16m39532, 2016.

[2] 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,” J. Broadcast Eng., vol. 25, no. 7, pp. 1073–1080, 2020.

[3] J.-B. Jeong, S. Lee, I. Kim, and E.-S. Ryu, “Implementing viewport tile extractor for viewport-adaptive 360-degree video tiled streaming,” in Proc. Int. Conf. Inf. Netw. (ICFIN), Jan. 2021, pp. 8–12.

[4] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou, “An overview of tiles in HEVC,” IEEE J. Sel. Topics Signal Process., vol. 7, no. 6, pp. 960–977, Dec. 2013.

[5] J. Son, D. Jung, and E.-S. Ryu, “Implementing motion-constrained tile and viewport extraction for VR streaming,” in Proc. 28th ACM SIGMM Workshop Netw. Operating Syst. Support Digit. Audio Video, Jun. 2018, pp. 61–66.

[6] J. M. Boyce, R. Dore, A. Dziembroski, J. Fleureau, J. Jung, B. Kroon, B. Salahieh, V. K. M. Vadakital, and L. Yu, “MPEG immersive video coding standard,” Proc. IEEE, vol. 109, no. 9, pp. 1521–1536, Sep. 2021.

[7] M. M. Hannuksela, Y. Yan, X. Huang, and H. Li, “Overview of the multiview high efficiency video coding (MV-HEVC) standard,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2015, pp. 2154–2158.

[8] S. Lee, J.-B. Jeong, I. Kim, S. Lee, and E.-S. Ryu, “Screen content coding analysis to improve coding efficiency for immersive video,” J. Broadcast Eng., vol. 25, no. 6, pp. 911–921, 2020.

[9] 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 Proc. 2018 ACM Int. Conf. Multimedia, 2020, pp. 3687–3695.

[10] D. Mieloch, A. Dziembowski, and M. Domanski, “Depth map refinement for immersive video,” IEEE Access, vol. 9, pp. 10778–10788, 2021.

[11] S. Fachada, D. Bonatto, Y. Xie, P. R. Alfase, M. Teratani, and G. Lafruit, “Depth image-based rendering of non-Lambertian content in MPEG immersive video,” in Proc. Int. Conf. 3D Immersion (IC3D), Dec. 2021, pp. 1–6.

[12] D. Park, S.-G. Lim, K.-J. Oh, G. Lee, and J.-G. Kim, “Nonlinear depth quantization using piecewise linear scaling for immersive video coding,” IEEE Access, vol. 10, pp. 4483–4494, 2022.

[13] S. Lee, J.-B. Jeong, and E.-S. Ryu, “Group-based adaptive rendering for 6DoF immersive video streaming,” J. Broadcast Eng., vol. 27, no. 2, pp. 216–227, 2022.

[14] J.-B. Jeong, S. Lee, and E.-S. Ryu, “Rethinking fatigue-aware 6DoF immersive video streaming: Focusing on MPEG immersive video,” in Proc. Int. Conf. Inf. Netw. (ICOFIN), Jan. 2022, pp. 304–309.

[15] L. Kondrad, V. K. M. Vadakital, and L. Ilola, CE-1.3: Frame Packed Video Sub-Bitstream Type, Standard ISO/IEC JTC1/SC29/WG11, IEEE/SM45274, 2020.

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

[17] H. Yuan, S. Kwong, X. Wang, W. Gao, and Y. Zhang, “Rate distortion optimized inter-view frame level bit allocation method for MV-HEVC,” IEEE Trans. Multimedia, vol. 17, no. 12, pp. 2134–2146, Dec. 2015.

[18] S. Heo, G. Bang, and G. H. Park, “Reusable HEVC design in 3D-HEVC,” ETRI J., vol. 38, no. 5, pp. 818–828, Oct. 2016.

[19] H. Amirpour, A. Pinheiro, M. Pereira, F. J. P. Lopes, and M. Ghanbari, “Efficient light field image compression with enhanced random access,” ACM Trans. Multimedia Comput., Commun., Appl., vol. 18, no. 2, pp. 1–18, May 2022.

[20] Call for Proposals on 3DoF+ Visual, Standard ISO/IEC JTC1/SC29/WG11, 125th MPEG meeting, MPEG/in18145, 2019.

[21] B. Kroon and B. Sonneveldt, Philips Response to 3DoF+ Visual CfP, Standard ISO/IEC JTC1/SC29/WG11, 126th MPEG Meeting, MPEG2019/m47179, 2019.

[22] V. K. M. Vadakital, K. Roimela, L. Ilola, J. Keränen, M. Pesonen, S. Schwarz, J. Lainema, and M. Hannuksela, Description of Nokian’s Response to CfP for 3DoF+ Visual Standard ISO/IEC JTC1/SC29/WG11, 126th MPEG meeting, MPEG2019/m47372, 2019.

[23] M. Domanski, A. Dziembowski, D. Mieloch, O. Stankiewicz, J. Stankowski, A. Grzela, G. Lee, and J. Sect, Technical Description of Proposal for Call for Proposals on 3DoF+ Visual Prepared by Poznan University of Technology (PUT) and Electronics and Telecommunications Research Institute (ETRI), Standard ISO/IEC JTC1/SC29/WG11, 126th MPEG Meeting, MPEG2019/m47407, 2019.

[24] J. Fleureau, F. Thudor, R. Dore, B. Salahieh, M. Dmytrychenko, and J. Boyce, Echnicolor-Intel Response to 3DoF+ CfP, Standard ISO/IEC JTC1/SC29/WG11, 126th MPEG Meeting, MPEG2019/m47445, 2019.

[25] B. Wang, Y. Sun, and L. Yu, Description of Zhejiang University’s Response to 3DoF+ Visual CfP, Standard ISO/IEC JTC1/SC29/WG11, 126th MPEG meeting, MPEG2019/m47684, 2019.
[26] J. Jung and B. Kroon, *Common Test Conditions for MPEG Immersive Video*, Standard ISO/IEC JTC1/SC29/WG4, MPEG/n0169, 2022.

[27] A. Wieckowski, J. Brandenburg, T. Hinz, C. Bartnik, V. George, G. Hege, C. Helmrich, A. Henkel, C. Lehmann, C. Stoffers, I. Zupanic, B. Bross, and D. Marpe, “V5ecC: An open and optimized VVC encoder implementation,” in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2021, pp. 1–2.

[28] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohn, “Overview of the versatile video coding (VVC) standard and its applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[29] M. Vutanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vainio, and T. D. Hämmäläinen, “Kvazar: Open-source HEVC/H.265 encoder,” in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 1179–1182.

[30] J.-B. Jeong, S. Lee, and E.-S. Ryu, “DWS-BEAM: Decoder-wise sub-picture bitmask extracting and merging for MPEG immersive video,” in *Proc. Int. Conf. Vis. Commun. Image Process. (VCIP)*, 2021, pp. 1–5.

[31] A. Hallapuro and M. M. Hannuksela, AHG3/AHG12: Subpicture Merging Software, document ISO/IEC JTC1/SC29/WG11, ITU-T SG 16 WP3, JVET-S0162, 2020.

[32] *Versatile Video Encoder (VVenC)* v1.4.0. Accessed: Aug. 1, 2022. [Online]. Available: https://github.com/fraunhoferhhi/vvenc/tree/v1.4.0

[33] A. Wieckowski, G. Hege, C. Bartnik, C. Lehmann, C. Stoffers, B. Bross, and D. Marpe, “Towards a live software decoder implementation for the upcoming versatile video coding (VVC) codec,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3124–3128.

[34] A. Dziembowski, D. Mieloch, J. Stankowski, and A. Grzelka, “IV-PSNR—the objective quality metric for immersive video applications,” *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jun. 1, 2022, doi: 10.1109/TCVT.2022.3179575.

[35] X. Xu, S. Liu, T.-D. Chuang, Y.-W. Huang, S.-M. Lei, K. Rapaka, C. Pang, V. Seregin, Y.-K. Wang, and M. Karczewicz, “Intra block copy in HEVC screen content coding extensions,” *IEEE J. Emerg. Sel. Topic Circuits Syst.*, vol. 6, no. 4, pp. 409–419, Dec. 2016.

[36] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, “HEVC-compliant tile-based streaming of panoramic video for virtual reality applications,” in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 601–605.

[37] F. Bossen, K. Sühring, A. Wieckowski, and S. Liu, “VVC complexity and software implementation analysis,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3765–3778, Oct. 2021.

[38] S. Lee, D. Jang, J. Jeong, and E.-S. Ryu, “Motion-constrained tile set based 360-degree video streaming using saliency map prediction,” in *Proc. 29th ACM Workshop Netw. Operating Syst. Support Digit. Audio Video (NOSSDAV)*, 2019, pp. 20–24.

[39] M. Santamaria, V. K. M. Vadakital, L. Kondrat, A. Hallapuro, and M. M. Hannuksela, “Coding of volumetric content with MIV using VVC subpictures,” in *Proc. IEEE 23rd Int. Workshop Multimedia Signal Process. (MMSys)*, Oct. 2021, pp. 1–6.

**SOONBIN LEE** (Graduate Student Member, IEEE) received the B.S. degree from Gachon University, in February 2020, and the M.S. degree from Sungkyunkwan University (SKKU), in February 2022, where he is currently pursuing the Ph.D. degree with the Department of Computer Science Education. Since July 2022, he has been a Visiting Researcher at the Multimedia Communications Group, Fraunhofer Heinrich Hertz Institute (HHI). His current research interests include video compression standards, MPEG immersive video (MIV), and deep learning based video coding.

**JONG-BEOM JEONG** (Graduate Student Member, IEEE) received the B.S. degree from Gachon University, in August 2018. He is currently pursuing the Ph.D. degree with the Department of Computer Science Education, Sungkyunkwan University (SKKU). From January 2020 to March 2020, he was a Visiting Student at the Department of Media Arts and Technology, University of California, Santa Barbara (UCSB), Santa Barbara, CA, USA. From August 2021 to January 2022, he was a Visiting Student at the Department of Computer and Information Technology, Purdue University, West Lafayette, IN, USA. His current research interests include video compression standards, MPEG immersive video (MIV), and deep learning based video coding.

**EUN-SEOK RYU** (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Korea University, Seoul, South Korea, in 1999, 2001, and 2008, respectively. In 2008, he was a Research Professor at the Research Institute for Information and Communication Technology, Korea University. From September 2008 to December 2010, he was a full-time Visiting Research Scientist II at the Georgia Centers for Advanced Telecommunications Technology (GCATT), School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. From January 2011 to February 2014, he was a Staff Engineer at the InterDigital Laboratories, San Diego, CA, USA, where he researched and contributed to next generation video coding standards, such as HEVC and SHVC. From 2014 to 2015, he was a Principal Engineer (the Director) at Samsung Electronics, Suwon, South Korea, where he led a Multimedia Team. From 2015 to 2019, he was an Assistant Professor at the Department of Computer Engineering, Gachon University, Seongnam, South Korea. He is currently an Associate Professor at the Department of Computer Education, Sungkyunkwan University (SKKU), Seoul. His research interests include multimedia computing systems that includes video source coding and wireless mobile systems. He is a member of ACM.