

Efficient Atlas Coding Strategy using cropping for Object-based MPEG Immersive Video

JAIYOUNG OH*, XIN LI*, KWAN-JUNG OH**, GWANGSOON LEE**, EUEE S. JANG*

* Department of Computer Science, Hanyang University, Seoul, Republic of Korea

** Immersive Media Research Section, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea

owjps@hanyang.ac.kr, liumei2012@hanyang.ac.kr, kjoh@etri.re.kr, gslee@etri.re.kr, esjang@hanyang.ac.kr

Abstract— In this study, we propose an efficient atlas coding strategy for the object-based MPEG Immersive Video (MIV) coding standard. The main concept of Object-based MIV is to use a video codec, such as High Efficiency Video Coding (HEVC) or Versatile Video Coding (VVC), to compress the 2D atlas videos generated from objects of interest videos taken through multiple cameras located in multiple positions in 3D space. However, the current coding method on the atlas in MIV is not effective for coding the objects of interest in videos because there are some cases where the atlas has an unnecessary area generated from the MIV encoder. If there is such an area that is not used in the atlas, the amount of computation required for coding can be reduced by decreasing the atlas resolution during encoding. The proposed method shows that the weighted-to-spherically uniform PSNR (WS-PSNR) based Bjøntegaard delta rate gains between -8.2% and -10.1%, and the Immersive Video PSNR (IV-PSNR) based on the Bjøntegaard delta rate gains between -5.7% and -7.1% compared to the reference model of MIV.

Keywords— Immersive video coding, MPEG immersive video, 6DoF video coding, Object-based coding, Metaverse, versatile video coding, high efficiency video coding

I. INTRODUCTION

Virtual reality (VR) and augmented reality (AR) are key technology areas in metaverse applications. The field of research to support VR/AR in the metaverse is immersive media where the user can experience three or six degrees of freedom (DOF). This capability provides users with a more realistic experience of viewing sports events, education and training, video conferences, and virtual tourism.

With the goal of providing use cases using immersive media, 360-degree video is a technology that allows users to view videos in all directions. However, it only supports three degrees of freedom (3DoF), meaning that it does not support motion parallax when an object moves relative to the viewer or when the viewer moves relative to an object in a fixed position [1], [2]. This limitation calls for a 6DoF immersive video service in which users can fully immerse themselves [1]. Immersive video that supports 6DoF can provide users with an experience that is similar to seeing the surroundings in the real world, as it supports motion parallax and considers changes in the viewer's position [1].

The Moving Picture Experts Group (MPEG) started standardization for coding immersive videos and developed the MPEG Immersive Video (MIV) coding standard (ISO/IEC 23090-12). The standard compresses videos of real or virtual 3D scenes using multiple cameras within a viewing space [1], [3]. The standardization of MIV has undergone several updates with a new version of the MIV incorporating new

technologies adopted through MPEG meetings. MIV reached the International Standard (IS), and further standardization efforts are underway to develop MIV edition 2 [4], [5].

Among the new use cases mentioned for developing MIV Edition 2 in MPEG, composition and compression for object-based MIV content were discussed [5]. Significant compression efficiency, real-time encoding and decoding performance are required to allow users to experience high-quality 6DoF contents [5]. This quality is essential because the 6DoF video allows dozens of views and depths can be acquired simultaneously from multiple locations in the viewing space [6].

Researchers have studied methods for object-based MIV. One study [7] demonstrated an object-based MIV that enables the encoding and selective rendering for objects of interest. However, the complexity of object-based coding needs to be reduced because it results in excessive weight in metaverse scenarios. The capacity of the metaverse is important for delivering high-quality videos to end users with a limited bandwidth. This limitation of 6DoF video consists of dozens of videos and depths acquired simultaneously from multiple locations in the viewing space [5]. If object-based coding is performed, the coding time and complexity would be further increased because numerous videos are required to compose a 6DoF video, and object units have to be processed.

This paper presents an enhanced approach to reduce this complexity and improve the coding efficiency of object-based MIV coding. If coding is performed on an object basis, there is almost no need to use all areas within the atlas video output after running both the MIV encoder and decoder. The complexity can be reduced by coding only the necessary area corresponding to the object. In addition, the bitrate can be reduced when compressing the atlas video with only the necessary areas corresponding to the object. The proposed method crops the atlas video to fit the packed patch by utilizing the patch information in the MIV encoder. The proposed method can enhance the end-to-end coding efficiency by minimizing unnecessary areas in the atlas video. The proposed method can support high-quality object-based 6DoF content to provide a hyper-real metaverse experience.

The remainder of this paper is organized as follows. In Section II, we introduce a basic description of the MIV codec and the process of object-based coding in the TMIV encoder. The patch packing process in the TMIV encoder and the limitations of the processing atlas are also explained. The proposed atlas coding strategy using a cropping algorithm is described in Section III. In Section IV, we evaluate the

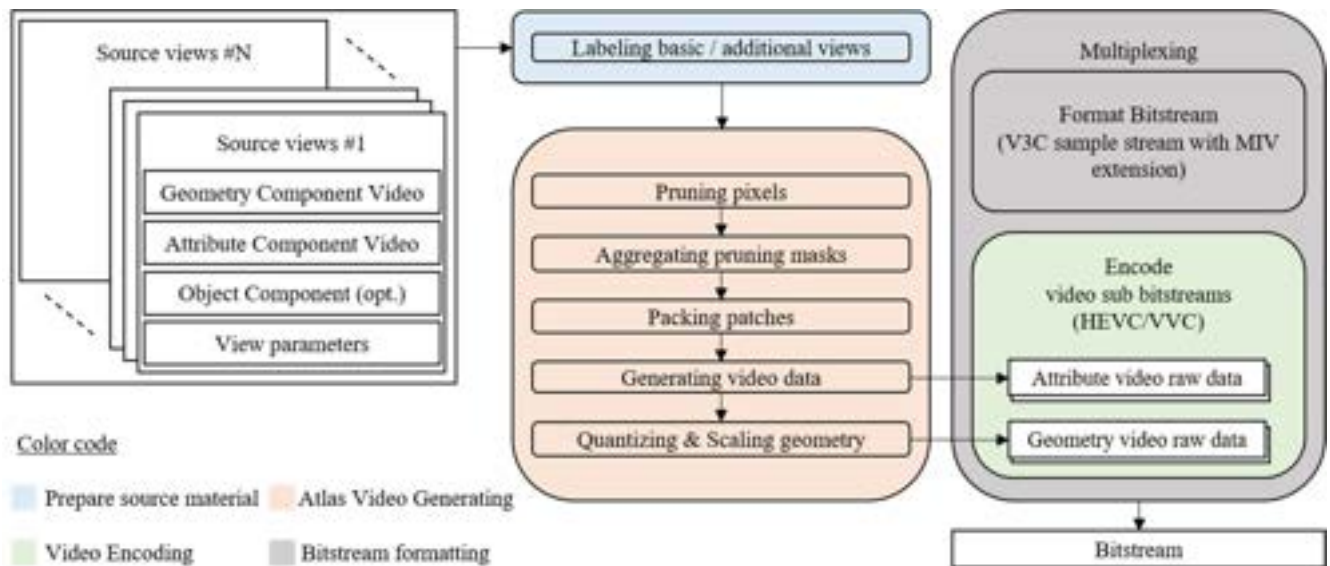


Figure 1. Overall architecture of the TMIV Encoder .

proposed method under the conditions of `miv_main_anchor` and entity-based. Finally, we conclude the paper in Section V.

II. BACKGROUND

A. Encoding process in MIV

Figure 1 shows the overall architecture of the MIV encoder including multiplexing and video compression. In the process of preparing the source material, the MIV encoder selects basic and additional views such that the redundancy of the selected basic views is spatially minimized. The basic views are selected based on the partitioning around medoids (PAM) algorithm and camera arrangement [8], [9]. The number of basic views is determined by calculations using the encoder configuration. Views other than the basic view are counted as additional views.

The process of pruning pixels involves measuring the interview redundancy in additional views, resulting in pruning masks that indicate the pixels that should be preserved [10]. In the process of generating an atlas video, the views labeled as basic views are preserved without pruning. On the other hand, the additional views are pruned by considering their spatial and visual overlap with the basic view and views that have already been pruned. Pruning pixels from these views involves projecting the basic view or pruned views onto the additional views in a hierarchical order based on the pruning graph. Unpruned pixels in the additional view are stored in a pruning mask [1]. The pruning mask affects the formation of the atlas video by accumulating over a range of consecutive video frames within an intra-period, thereby efficiently encoding the atlas video. The number of accumulated pixels in the pruning mask increases, particularly in regions with motion [9]. Because the pruned pixels are subsequently stored in a limited number of atlas videos through several processes, not all the increased pruning pixels are likely to be stored in the atlas video in cases of dynamic motion within regions.

Patches are formed through clustering, which involves combining pixels into a pruning mask [10]. A cluster is a set of connected pixels within the mask, and each cluster has a bounding box around it. To compress the atlas video more efficiently, it is advisable to reduce invalid pixels by merging or splitting the clusters within the bounding box. Patches

formed through clustering are packed into the atlas using the MaxRect algorithm [11]. Patches are rotated or flipped to optimize packing efficiency and are packed into the atlas in the order of decreasing size. After the patch packing process, the geometry atlas video data and attribute atlas video data are generated, as shown in figure 2. The geometry atlas video data is then scaled and quantized.

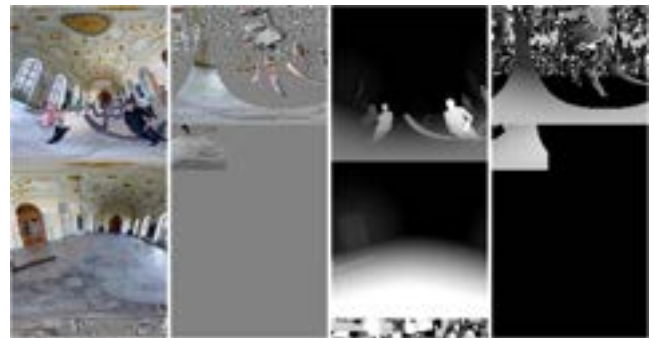


Figure 2. Atlases videos generated by MIV encoder: 2 texture videos and 2 geometry videos of Museum sequence.

After the atlas video is generated by the MIV encoder, the attribute and geometry video raw data are encoded using a 2-D video codec, such as HEVC or VVC. During the bitstream formatting process, the bitstream containing the metadata generated by the MIV encoder and compressed bitstreams from the 2-D video codec are multiplexed into a single bitstream. This process creates a single bitstream that is suitable for storage, transmission, and MIV decoding.

B. Encoding Processes of entity-based coding in MIV

Figure 3 shows the architecture of an entity-based MIV encoder. The entity-based MIV encoder features an object separator and entity-based atlas constructor. Entity-based coding requires an object component that indicates the objects corresponding to the pixels within the input content. In the entity-based MIV encoder, the object components are inputted along with the geometry component, attribute component, and view parameters. Figure 4 shows an input source in the entity-based MIV encoder, where objects in the video are shown with colors applied to make them easier to see. The entity-based MIV encoder labels the basic and additional views within the

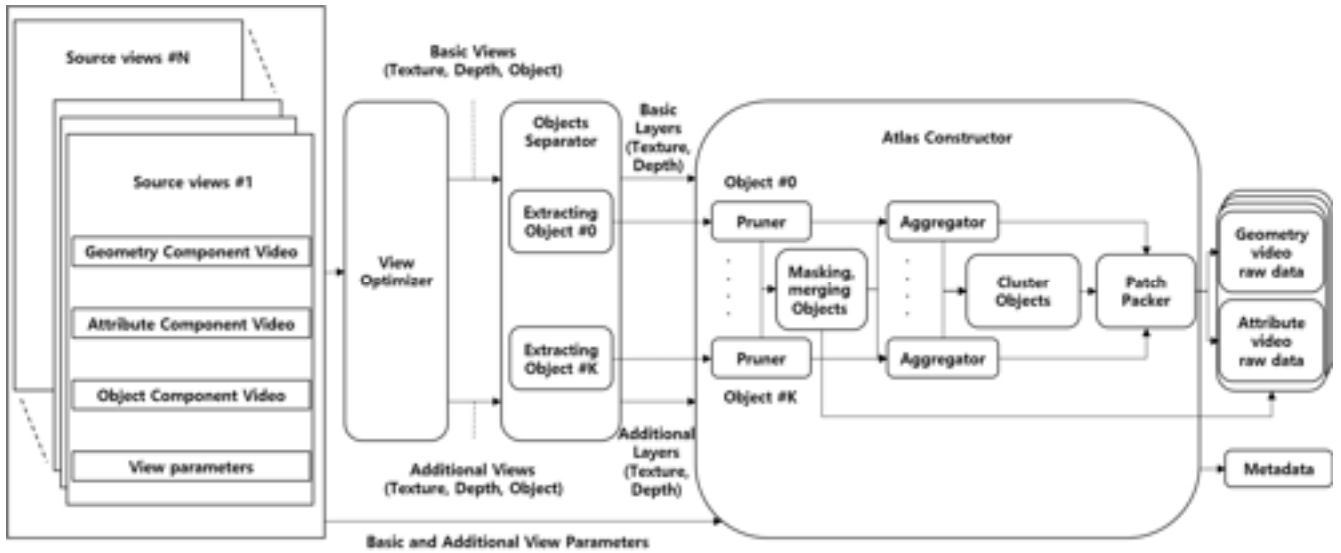


Figure 3. Overall architecture of the entity-based MIV Encoder.

input source views. However, the basic views in the entity-based MIV encoder are used only for pruning and packing, and not for streaming all basic view data. This is because each patch only carries pixels that belong to its own object [7].

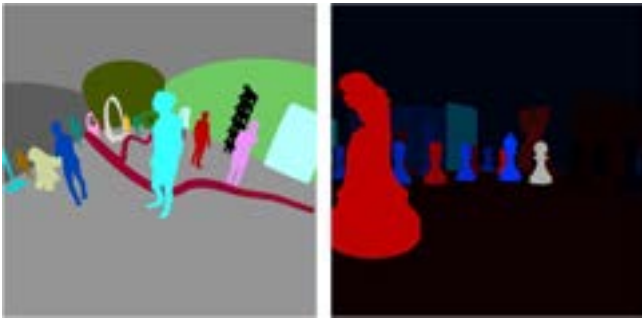


Figure 4. Object component videos used as the input source of entity-based MIV encoder: Museum sequence (left) and Chess Pieces sequence (right).

Object layers, including basic and additional layers, are extracted from geometry component videos and attribute component videos using object component videos. Basic masks are updated in the object masker using basic layers, and pruning masks are generated through a pruner that measures the interview redundancy object by object in the additional layers based on basic layers [7]. The object masks consist of ObjectID, which indicates that objects are merged with the basic masks. This allows the ObjectID to be tracked and the patches can be tagged with the right ObjectID [7]. The pruning masks and merged object masks are aggregated by the aggregator. The aggregator then finishes the accumulation of the masks after processing all frames within an intra-period [7]. Clustering is performed for each object, and the clusters are tagged with the related objectIDs and queued together [7]. Patches generated from clusters are packed into the atlas to create raw attribute and geometry video data.

C. Patch packing into Atlas in MIV

Patches generated from clustering are packed into atlases to create compact atlas videos to improve coding efficiency. All patches are sorted in decreasing order before patch packing [1], [9]. Then, the MaxRect algorithm [11] is used in the encoder to sequentially pack the patches, offering the advantages of high packing efficiency and speed. Patch

packing in the MIV encoder can provide options for the patches to be rotated 90°, 180° and 270°, or for the patches to be flipped vertically according to the given syntax [1]. The algorithm that packs patches starting from the top-left corner of the free space within the atlas is used in the reference software of the MIV encoder. There are occupied regions in a patch, and the patches are packed so as not to invade each other's occupied regions [12]. As patches are packed in order of size, free space exists between the occupied regions of the packed patches. If there is a patch that can be packed in free space between occupied regions, the patch can be packed. The MIV decoder must support the process of patch packing in reverse. Therefore, the MIV bitstream includes metadata such as the packing order, position and size of each patch, and the source view index from which the patch is extracted [1], [9].

D. Limitations in calculating the parameters of atlas in MIV

In the initial stage, the frame size and the number of atlases are calculated using source view information and configurations. The configurations such as the maximum size of a picture in luma samples, the maximum luma sample rate and the maximum number of atlases are also included in the calculation of the parameters of the atlas. The width of the atlas frame is typically set to the widest source view [1]. And the height of the atlas frame is set as large as possible within the range of value calculated using configuration and the width of the atlas frame. The number of atlases is calculated using the configuration, and it does not exceed the maximum number of atlases set in configuration. The parameters of atlas are calculated before generating the atlas video, and the generation of atlas video is processed in Group of Pictures (GOP) units based on the calculated atlas parameters. However, the calculation of atlas parameters is not processed in GOP units. Therefore, adjusting the frame size and the number of atlas videos differently in GOP units has not yet been supported.

III. CROPPING ALGORITHM FOR ATLAS CODING BASED ON PATCH PACKING IN MIV

The proposed cropping algorithm for atlas coding is based on the patch packing algorithm used in the MIV. Patch packing in MIV places the patches into an atlas without

overlapping the occupied regions. Each patch can be rotated or flipped for proper placement in an atlas. Then, there are patch lists for each atlas with information that indexes the upper left position and sizes of the patch. The proposed method is designed to reduce the resolution of the video called atlas (Figure 2) generated by MIV encoder. The proposed method minimizes the resolution of the atlas videos using patch information that indexes the upper left position and size of each patch's width and height. The atlas videos can be cropped to videos with a minimal bounding box by using the patch information shown in figure 5.

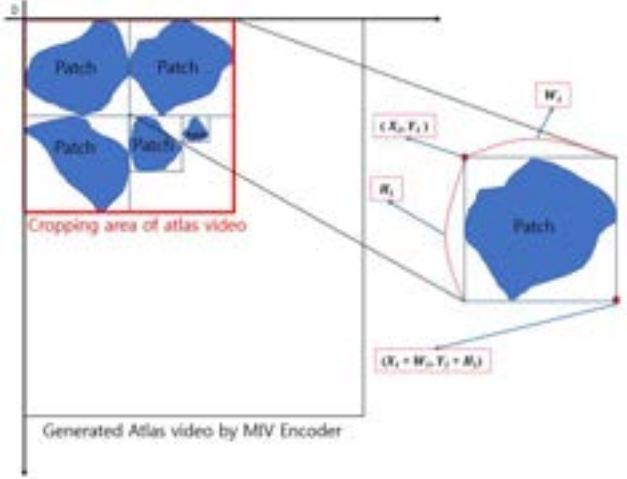


Figure 5. Proposed atlas cropping in MIV encoder using patch information

Each patch packed within the atlas has a rectangular shape. Each patch has its own coordinates for the upper-left point, horizontal length, and vertical length. In figure 5, which shows the i -th patch being packed, X_i represents the x coordinate of the upper left point of the patch, and Y_i represents the y coordinate of the upper left point of the patch. A value called W_i represents the horizontal length of the patch, and a value called H_i represents the vertical length of the patch. The coordinates of the bottom-right point of patch (B) can be calculated using patch information as follows:

$$X_i^B = X_i + W_i \quad (1)$$

$$Y_i^B = Y_i + H_i \quad (2)$$

where X_i is the top left x coordinate of i -th patch, Y_i is the bottom left of y coordinate of i -th patch, W_i is the width of i -th patch, H_i is the height of i -th patch.

In the TMIV reference encoder [10], patches are packed in order from the largest to the smallest within the atlas frame. The patches are packed from the upper left corner of the atlas frame, and patch packing occurs until there is no valid region within the atlas frame. Each patch packed in the atlas has its own coordinates of the bottom right side of the patch. Among the x and y coordinates of the bottom right corner of the patch for each patch, the x and y coordinates furthest from the upper left corner of the atlas can be calculated. The x coordinate furthest from the upper left corner of the atlas can be set as the horizontal length of the cropping atlas. The y coordinate furthest from the upper left corner of the atlas can be set as the vertical length of the cropping atlas. The horizontal and vertical lengths of the cropping atlas (CA) are selected from the maximum value among the coordinates as follows:

$$CA_x = \max(X_i^B) \quad (3)$$

$$CA_y = \max(Y_i^B) \quad (4)$$

where X_i^B represents the bottom-right x coordinate of the i -th patch, and Y_i^B represents the bottom-right y coordinate of the i -th patch. We select the maximum values of X_i^B and Y_i^B among all the patches as CA_x and CA_y , respectively.

When cropping an atlas, the complexity of the proposed method, which uses patch list information that indexes the position and size of each patch, is lower than that of the method that calculates the valid pixels of patches in the atlas after the patch packing is completed. The proposed method simply searches for position and size information for at least one to hundreds or thousands of patches. However, to determine the cropped atlas size for an atlas in which the process of patch packing is completed, the valid pixels of the patches must be examined within a number of pixels corresponding to the size multiplied by the width and height of the atlas. For example, if the atlas has a height of 4096 and a width of 2048, the amount of 8,388,608 ($=4096 \times 2048$) pixels need to be checked to determine whether each pixel corresponds to the patch or not. In this process, the number of pixels that must be examined is often hundreds of thousands or more. Therefore, it can be proposed that the method using the position and size information of each patch in the patch list is an effective method for cropping the atlas in the MIV encoder.

After cropping the atlas videos, the resolution of each atlas video may vary from frame to frame in some cases. This is because the resolution of the bounding box may vary depending on the result of the packed patches within the atlas, on a frame-by-frame basis. The resolution of the atlas with the minimum bounding box is calculated to contain all patches within a GOP unit. A cumulative computing process is applied per frame to obtain the minimum width and height of the bounding box within a GOP unit. The width and height of the bounding box surrounding all packed patches are used as the cropping area size for the atlas video. The resolution of the cropping atlas was set to a multiple of 128, considering the CTU (Coding Tree Unit) size set in the VVenC configuration.

IV. EXPERIMENTAL RESULTS

Experiments are conducted to evaluate the proposed cropping algorithm for atlas coding in MIV. The cropping algorithm is implemented on top of the patch-packing process in the TMIV. The proposed method is evaluated under common test conditions (CTC) [13] of the MIV in the configuration of the MIV main anchor. The proposed method is also evaluated for the configuration of entity-based coding. The test conditions for the experiments and their results are described below.

A. EXPERIMENTAL CONDITIONS

In this study, two experiments were conducted. First, we conducted a comparative experiment using the original TMIV and the proposed method in the TMIV with the MIV main anchor configuration. When testing the MIV reference software, the experiment is conducted based on the MIV main anchor. The experiment is conducted using CTCs to fairly compare the result of the experiment.

Second, we compared the coding efficiency of each object using an entity-based coding configuration. Among the

datasets of several sequences listed in Table 1, only the museum sequence was used. This is because the object map video needs to use an entity-based coding configuration, and the museum sequence provides the necessary information related to the object map video. However, to conduct experiments by coding each object, two modifications are needed from the existing Peak Signal-to-Noise Ratio (PSNR) measurement method. First, the original source files separated for each object are required to measure the PSNR for each object. An object map is required to create the original files separated by an object. It is possible to separate objects using object ID information in the object map. Second, when measuring the PSNR, some objects may not exist in a specific view. The PSNR in MIV measures the PSNR for multiple views that are defined in the dataset and CTC. However, when PSNR is measured by coding for each object, the results are identical to the original file that is separated by the object for views where no object exists. Therefore, the PSNR must be measured by excluding specific views in which no object exists for each object. This makes it possible to accurately measure the PSNR for each object.

Anchor is defined as the MIV main anchor in CTCs. Because the definition of the TMIV main anchor is coded with TMIV and VVenC, some views are fully packed while taking only patches of others [13]. The atlases generated by the TMIV Encoder are compressed using the VVC open-source software called VVenC [14]. As shown in Table 1, five rate points for texture in VVenC are considered for the MIV main anchor. One single-geometry QP corresponds to each texture QP. The mapping from the texture QP_t shown in Table 1 to the geometry QP_g is calculated as follows:

$$QP_g = \max(1, [-14.2 + 0.8 QP_t]) \quad (5)[13]$$

TABLE I. VARIOUS TEST SEQUENCES AND LISTS OF TEXTURE QPs FOR EACH TEST SEQUENCE IN MIV MAIN ANCHOR [13]

Sequence	QP1	QP2	QP3	QP4	QP5
Classroom Video	26	28	33	41	51
Museum	29	38	44	48	51
Fan	30	35	41	46	51
Kitchen	17	24	29	35	41
Chess	17	25	31	37	45
Group	22	28	34	39	46

An objective evaluation is performed by comparing all synthesized views generated by the TMIV decoder between the MIV main anchor and the proposed method. The objective quality is evaluated using the software tool called IV-PSNR [15]. The IV-PSNR software computes not only the IV-PSNR score, but also the WS-PSNR for both perspective and omnidirectional content. Additionally, the BD-rate and BD-PSNR values can be calculated using the IV-PSNR scores, WS-PSNR scores, and bitrate of the compressed video, considering all source views.

B. EXPERIMENTAL RESULTS

Table 2 lists the BDBR difference between the MIV main anchor and the proposed method for each sequence. A ‘high’ BD-rate is the BD-rate calculated from QP1 to QP4 among the five QPs, while the ‘low’ BD-rate is calculated from QP2 to QP5. The average coding gain on Y-PSNR ranges from -0.0% to -0.1%. The average coding gain on WS-PSNR ranges between 0.1% and -0.0%. In the objective quality evaluation

using Y-PSNR and IV-PSNR, it is observed that there is almost no difference in the quality and bitrate between the MIV main anchor and the proposed method.

TABLE II. HIGH/LOW BD-RATE OF Y-PSNR AND IV-PSNR ON 17 FRAMES BETWEEN THE MIV MAIN ANCHOR AND THE PROPOSED METHOD FOR EACH SEQUENCE.

Sequence	High-BR BD rate	Low-BR BD rate	High-BR BD rate	Low-BR BD rate
	Y-PSNR	Y-PSNR	IV-PSNR	IV-PSNR
ClassroomVideo	-0.5%	-0.6%	-0.4%	-0.6%
Museum	-0.1%	-0.2%	-0.1%	-0.2%
Fan	0.7%	0.6%	0.8%	0.6%
Kitchen	-0.3%	-0.2%	-0.1%	-0.2%
Chess	0.0%	0.0%	0.0%	0.0%
Group	0.0%	-0.1%	0.2%	0.1%
Average	-0.0%	-0.1%	0.1%	-0.0%

Table 3 presents a resolution comparison between the uncropped and cropped atlas videos. Experiments were conducted using two atlases (c00 and c01) for geometry and texture. When encoding in TMIV with the MIV main anchor configuration, patches are packed into the first atlas (c00) to the extent that space is available for patch packing. The c00 atlas does not appear cropped. However, in the case of the c01 atlas, cropping occurs because there is an unused space where patches are not packed. As seen in Table 3, it can be observed through the reduction ratio that the cropping mainly occurs in C01.

TABLE III. RESOLUTION COMPARISON BETWEEN THE UNCROPPED AND CROPPED ATLAS VIDEOS OF 17 FRAMES FOR EACH SEQUENCE.

Sequence		Uncropped C00 atlas resolution	Cropped C00 atlas resolution	Reduction ratio of C00 atlas resolution after cropping (%)	Uncropped C01 atlas resolution	Cropped C01 atlas resolution	Reduction ratio of C01 atlas resolution after cropping (%)
ClassroomVideo	geo atlas	2048x1088	2048x1088	0%	2048x1088	576x1088	72%
Museum		1024x2176	1024x2176	0%	1024x2176	1024x1088	50%
Fan		960x2320	960x2320	0%	960x2320	960x832	64%
Kitchen		960x2320	960x2320	0%	960x2320	960x1216	48%
Chess		1024x2176	1024x2176	0%	1024x2176	1024x2176	0%
Group		960x2320	960x2320	0%	960x2320	960x1792	23%
ClassroomVideo	tex atlas	4096x2176	4096x2176	0%	4096x2176	1152x2176	72%
Museum		2048x4352	2048x4352	0%	2048x4352	2048x2176	50%
Fan		1920x4640	1920x4640	0%	1920x4640	1920x1664	64%
Kitchen		1920x4640	1920x4640	0%	1920x4640	1920x2432	48%
Chess		2048x4352	2048x4352	0%	2048x4352	2048x4352	0%
Group		1920x4640	1920x4640	0%	1920x4640	1920x3584	23%

Table 4 presents a resolution comparison between uncropped and cropped atlas videos for each entity. It shows the extent of cropping for each entity when coding each object using the object map in the museum dataset. Entities 0 to 5 represent objects such as walls or ceilings, while entities 6 to 24 represent objects such as people, animals, and sculptures. The resolution of the cropped atlas varies depending on how the patches are packed within the atlas for each entity. The average resolution reduction ratio after cropping for entities 0 to 5 is 54%. The average resolution reduction ratio after cropping for entities 6 to 24 is 93%. In other words, on average more resolution reduction occurs after cropping for entities such as people, animals, and sculptures compared to entities such as walls and ceilings.

TABLE IV. RESOLUTION COMPARISON BETWEEN UNCROPPED AND CROPPED ATLAS VIDEOS OF 17 FRAMES FOR EACH ENTITY IN MUSEUM SEQUENCE.

Sequence	Entity	Uncropped pre atlas resolution	Cropped pre atlas resolution	Reduction ratio of pre atlas resolution after cropping (%)	Uncropped post atlas resolution	Cropped post atlas resolution	Reduction ratio of post atlas resolution after cropping (%)
Museum	0	1024x2176	1024x2176	0%	2048x4352	2048x4352	0%
	1	1024x832	1024x832	0%	2048x1664	2048x1664	0%
	2	1024x704	1024x704	0%	2048x1408	2048x1408	0%
	3	1024x384	1024x384	0%	2048x768	2048x768	0%
	4	448x520	448x520	0%	896x840	896x840	0%
	5	1024x1792	1024x1792	0%	2048x3584	2048x3584	0%
	6	384x256	384x256	0%	768x512	768x512	0%
	7	256x128	256x128	0%	512x256	512x256	0%
	8	256x192	256x192	0%	512x384	512x384	0%
	9	256x192	256x192	0%	512x384	512x384	0%
	10	1024x256	1024x256	0%	2048x512	2048x512	0%
	11	1024x1088	1024x1088	0%	2048x2176	2048x2176	0%
	12	1024x2176	640x576	85%	2048x4352	1280x1152	83%
	13	1024x192	1024x192	0%	2048x384	2048x384	0%
	14	1024x384	1024x384	0%	2048x768	2048x768	0%
	15	1024x192	1024x192	0%	2048x384	2048x384	0%
	16	384x128	384x128	0%	768x256	768x256	0%
	17	320x128	320x128	0%	640x256	640x256	0%
	18	320x128	320x128	0%	640x256	640x256	0%
	19	448x128	448x128	0%	896x256	896x256	0%
	20	640x128	640x128	0%	1280x256	1280x256	0%
	21	320x128	320x128	0%	640x256	640x256	0%
	22	256x128	256x128	0%	512x256	512x256	0%
	23	128x64	128x64	100%	256x128	256x128	100%
	24	192x64	192x64	99%	384x128	384x128	99%

Table 5 presents the BDBR differences between the entity-based coding anchor and the proposed method for each entity. The average coding gain in terms of Y-PSNR ranges from -5.7% to -7.1%, while the average coding gain in WS-PSNR ranges from -8.2% to -10.1%. From Tables 4 and 5, we can observe that coding efficiency increases as more areas of the atlas are cropped for each entity. It can be observed that a significant resolution reduction in the atlas video due to cropping tends to result in gains in the BD-rate.

TABLE V. HIGH AND LOW BD-RATES OF Y-PSNR AND IV-PSNR ON 17 FRAMES BETWEEN THE ENTITY-BASED CODING ANCHOR AND THE PROPOSED METHOD FOR EACH ENTITY IN THE MUSEUM DATASET.

Sequence	Entity	High-BR BD rate	Low-BR BD rate	High-BR BD rate	Low-BR BD rate
		Y-PSNR	Y-PSNR	IV-PSNR	IV-PSNR
Museum	0	0.0%	0.0%	0.0%	0.0%
	1	-0.2%	-0.5%	-0.0%	-0.4%
	2	0.1%	-0.7%	0.2%	-0.9%
	3	-1.2%	-0.7%	-1.2%	-0.8%
	4	-12.2%	-12.2%	-13.5%	-13.0%
	5	-0.3%	-0.4%	-0.3%	-0.5%
	6	-10.7%	-11.9%	-9.7%	-11.3%
	7	-10.7%	-10.9%	-10.2%	-10.7%
	8	-8.1%	-8.5%	-6.7%	-9.4%
	9	-6.6%	-7.5%	-6.7%	-7.4%
	10	-4.2%	-2.9%	-1.8%	-2.1%
	11	-6.6%	-4.6%	-5.8%	-3.1%
	12	-0.0%	-0.3%	-0.9%	-0.4%
	13	-1.3%	-0.5%	-3.2%	-3.2%
	14	-5.8%	-7.0%	-6.3%	-7.1%
	15	-0.7%	-2.4%	---	-17.7%
	16	-9.1%	-10.3%	-65.8%	-15.0%
	17	-4.4%	-8.4%	---	-79.2%
	18	-6.0%	-7.6%	-7.2%	-8.3%
	19	-9.4%	-9.5%	-8.7%	-9.2%
	20	-10.2%	-9.7%	-7.0%	-9.8%
	21	-11.7%	-12.8%	-12.4%	-13.3%
	22	-13.3%	-14.2%	-6.3%	-14.0%
	23	8.2%	-15.8%	-18.6%	-17.4%
	24	-17.7%	-18.8%	-13.7%	1.1%
	Average	-5.7%	-7.1%	-8.2%	-10.1%

Figure 6 displays snapshots of the atlas image frames from the anchor and proposed methods for entities 4, 7, 9, and 21 in the museum sequence. The first row corresponds to the anchor output, and the second row represents the output of the proposed method. As shown in figure 6, it is evident that the atlas resolution in the proposed method is reduced because of

cropping in the remaining areas where patches are not packed. In other words, when performing object-based coding, it can be subjectively observed that the patches corresponding to the objects actually occupy a small portion of the atlas area.



Figure 6. Anchor(first row) and proposal(second row) of atlas video for entity 4(first column), 7(second column), 9(third column), 21(fourth column) in museum sequence.

Figure 7 and figure 8 show the performance of the proposed method compared to the anchor in terms of WS-PSNR and IV-PSNR, respectively, for entities 4, 7, 9, and 21. The horizontal axis represents the bitrate (in Mbps), and the vertical axis represents the PSNR. Measurements were performed at five different points. The results show that the PSNR tends to be similar between the anchor and the proposal, with the proposal sometimes being slightly lower than the anchor. However, in terms of bitrate, the proposed method consistently shows a reduction compared to the anchor.

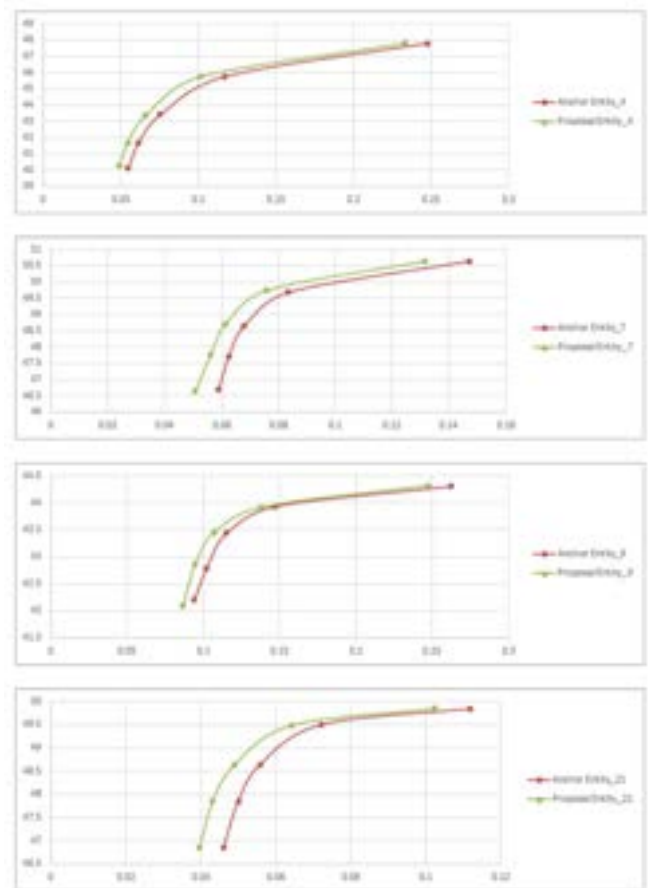


Figure 7. Entity-level WS-PSNR on 17 frames

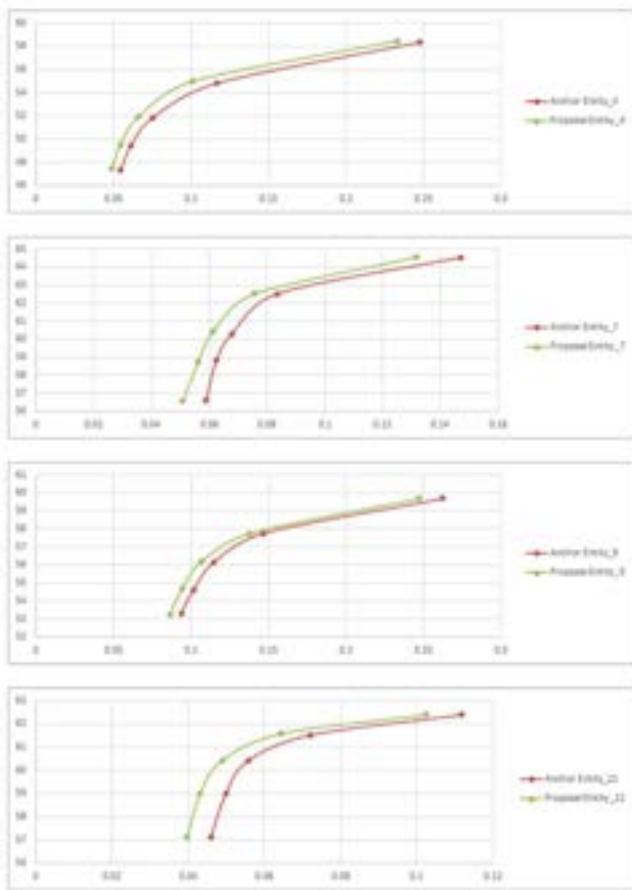


Figure 8. Entity-level IV-PSNR on 17 frames

Figure 9 shows examples of rendered viewports for entities 4, 7, 9, and 21 in the museum sequence for QP1. The left column displays the viewports rendered by the anchor, and the right column shows the viewports rendered using the proposed method. The first row corresponds to entity 4, the second to entity 7, the third to entity 9, and the fourth to entity 21. From figure 9, we can compare the subjective quality between the anchor and proposal for each object. Upon comparing the subjective quality of the anchor and proposal, no significant differences in the rendered results are apparent.



Figure 9. Rendered anchor viewports (Left) and rendered proposed viewports (right) of entity 4(first row), 7(second row), 9(third row), 21(fourth row) in the Museum sequence for QP1.

V. CONCLUSION

This study presents an efficient atlas coding strategy that uses cropping algorithms to improve the performance of object-based MIV. In the proposed method, the cropping algorithm leverages the patch information generated during the packing of patches in the MIV encoding. Using the position of the upper-left point, along with width and height information, the area of the atlas that requires cropping can be calculated after the patches are packed, excluding the area of the packed patches. The experimental results show that, overall, the larger the cropped area of the atlas, the greater the end-to-end coding gain. This is because a larger cropped area of the atlas results in a lower bitrate generated during encoding. The proposed method can provide an efficient coding strategy for scenarios in which object-by-object coding is necessary. It is particularly valuable in situations that require object-by-object processing within a scene in application fields such as VR and AR in the metaverse.

ACKNOWLEDGMENT

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00207, Immersive Media Research Laboratory).

REFERENCES

- [1] J. M. Boyce et al., "MPEG Immersive Video Coding Standard," in *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1521-1536, Sept. 2021, doi: 10.1109/JPROC.2021.3062590.
- [2] Y. Ye, J. M. Boyce and P. Hanhart, "Omnidirectional 360° Video Coding Technology in Responses to the Joint Call for Proposals on Video Compression With Capability Beyond HEVC," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1241-1252, May 2020, doi: 10.1109/TCSVT.2019.2953827.
- [3] R. Koenen and M. Champel, Requirements MPEG-I Phase 1b, document ISO/IEC JTC1/SC29/WG11 MPEG/N17331, Jan. 2018.
- [4] Information technology—Coded representation of immersive media—Part 12: MPEG immersive video, Standard ISO/IEC 23090-12:2023, Aug. 2023.
- [5] Draft Use Case and Requirements for MIV Edition-2, document ISO/IEC JTC1/SC29/WG2/N0157, Jan. 2022.
- [6] K. Müller, P. Merkle and T. Wiegand, "3-D Video Representation Using Depth Maps," in *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643-656, April 2011, doi: 10.1109/JPROC.2010.2091090.
- [7] Salahieh, Basel, Wayne Cochran, and Jill Boyce. "Delivering Object-Based Immersive Video Experiences." *Electronic Imaging 2021.18* (2021): 103-1.
- [8] L. Kaufman and P. J. Rousseeuw, *Partitioning Around Medoids (Program PAM)* (Wiley Series in Probability and Statistics). Hoboken, NJ, USA: Wiley, pp. 68–125.
- [9] V. K. M. Vadakital, A. Dziembowski, G. Lafruit, F. Thudor, G. Lee, and P. R. Alface, "The MPEG immersive video standard—Current status and future outlook," *IEEE Multimedia Mag.*, vol. 29, no. 3, pp. 101–111, Jul. 2022.
- [10] Test Model 18 for MPEG Immersive Video, document ISO/IEC JTC1/SC29/WG4 N0409, Oct. 2023.
- [11] J. Jylanki, "A Thousand Ways to Pack the Bin – A Practical Approach to Two-Dimensional Rectangle Bin Packing," retrieved from <http://pds25.egloos.com/pds/201504/21/98/RectangleBinPack.pdf>, Feb. 2010.
- [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] Common Test Conditions for MPEG Immersive Video, document ISO/IEC JTC1/SC29/WG4/N0232, Jul 2022.
- [14] Fraunhofer HHI VVenC Software Repository. Accessed: Jan 2024. [Online]. Available: <https://github.com/fraunhoferhhi/vvenc>
- [15] IV-PSNR Software Repository. Accessed: Jan 2024. [Online]. Available : <https://gitlab.com/mpeg-i-visual/ivpsnr>



JAIYOUNG OH received a B.S. degree from the Korea National University of Education, Korea. He is currently studying at Hanyang University as a combined MS/PhD student. His current research interest includes immersive video coding.



Xin Li received the B.S and M.S degrees in computer science and engineering from Kyungpook National University, South Korea, in 2012 and 2014, respectively. He was actively working on developing multimedia applications of 4D contents in a start-up company (2014). From 2017 to 2020, he was an assistant manager at the software R&D center of Genoray Co. Ltd., Korea, where he was involved in developing dental imaging software. He is currently a Ph. D. student at Hanyang University. He has authored 14 articles on MPEG standardization, and his research interests include computer graphics and multimedia compressions.



KWAN-JUNG OH received the B.S. degree in electronic computer engineering from Chonnam University, Gwangju, Republic of Korea, in 2002, and the M.S. and Ph.D. degrees in information and communications engineering from the Gwangju Institute of Science and Technology (GIST), Gwangju, in 2005 and 2010, respectively. From 2010 to 2013, he was with the Samsung Advanced Institute of Technology (SAIT), where he was involved in standardization activities of 3D video coding. He joined the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea, in 2013, where he is currently a Senior Researcher with the Broadcasting and Media Research Laboratory. His research interests include 3D image and video coding, immersive media, and holography.



GWANGSOON LEE received the Ph.D. degree in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2004. He joined the Electronics and Telecommunications Research Institute (ETRI), Daejeon, in 2001. He is currently a Principal Researcher with the Realistic-Media Research Sector. He has been involved in immersive video coding standards in ISO/IEC MPEG. His research interests include the immersive video processing, light-field imaging systems, and the three-dimensional video systems.



EUEE S. JANG (Senior Member, IEEE) received a B.S. degree from Jeonbuk National University, South Korea, and a Ph.D. degree from SUNY at Buffalo, Buffalo, NY, USA. He is currently a professor at the Department of Computer Science and Engineering, Hanyang University, Seoul, South Korea. He has authored more than 325 articles on MPEG standardization, 90 journal articles and conference papers, 65 pending or accepted patents, and two book chapters. His research interests include image and video coding, reconfigurable video coding, and computer graphics. He also received three ISO/IEC certificates of appreciation for his contributions to MPEG-4 development. Finally, he received the Presidential Award from the Korean government for his contribution to MPEG standardization.