Towards Viewport-dependent 6DoF 360 Video Tiled Streaming for Virtual Reality Systems

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

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

Il-Woong Ryu
Gachon University
Department of Computer Engineering
Seongnam, Republic of Korea
dlfdnd96@gc.gachon.ac.kr

Tuan Thanh Le
Gachon University
Department of Computer Engineering
Seongnam, Republic of Korea
tuanlt@gc.gachon.ac.kr

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

ABSTRACT

Previous studies of 360-degree video streaming with regard to virtual reality allowed users to move their head freely, while their position is fixed according to the camera's location in virtual reality. One of the approaches to overcome the problem is transmitting multi-view video to provide six degrees of freedom (6DoF). However, 6DoF streaming system implementation is challenging because multiple high-quality video streaming requires several decoders and a high bandwidth. Therefore, this paper proposes a viewport-dependent high-efficiency video coding (HEVC)-compliant tiled streaming system on test model for immersive video (TMIV), MPEG-Immersive multiview compression reference software. This paper proposes a 6DoF viewport tile selector (VTS) for multiple 360-degree video tiled streaming. Furthermore, this paper introduces a viewport-dependent multiple-tile extractor. The proposed system detects the user's head movement, selects the tile sets that correspond to the user's viewport, extracts tile bitstreams, and generates single bitstream. The extracted bitstream is transmitted and decoded to render the user's viewport. The proposed viewport-dependent streaming method can reduce the decoding time as well as the bandwidth. Experimental results demonstrated 12.04% bjontegaard delta rate (BD-rate) saving for the luma peak signal-to-noise ratio (PSNR) compared to those obtained via the TMIV anchor without tiled encoding and a 55.51% decoding time saving compared to those obtained via the TMIV anchor with the existing tiled streaming method.

CCS CONCEPTS

• Computing methodologies → Virtual reality; • Information systems → Multimedia streaming.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

MM '20, October 12–16, 2020, Seattle, WA, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7988-5/20/10...\$15.00

<https://doi.org/10.1145/3394171.3413712>

KEYWORDS

Virtual reality, 3DoF+, 6DoF, MPEG-I, Immersive video, MCTS, Viewport-dependent streaming

ACM Reference Format:

Jong-Beom Jeong, Soonbin Lee, Il-Woong Ryu, Tuan Thanh Le, and Eun-Seok Ryu. 2020. Towards Viewport-dependent 6DoF 360 Video Tiled Streaming for Virtual Reality Systems. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, October 12–16, 2020, Seattle, WA, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3394171.3413712>

1 INTRODUCTION

Currently, virtual reality (VR) technology has gained immense popularity and various head-mounted display (HMD) devices are available in the market. Moreover, the demand for graphical environment and realistic 360-degree scene representation on an HMD is increasing. HMD is a display device that a user can wear on the head and view videos; it has a 360-degree viewing range to achieve an immersive experience. Low latency and high-quality videos are required to increase the user's quality of experience (QoE). For example, Champel et al.[2] reported that a video with a 12K resolution, 90 fps, and 20 ms of motion-to-photon latency is required to reduce the motion sickness. To meet the challenging requirements, several approaches to overcome the aforementioned problems have been proposed. For example, because a user views only a portion of a 360-degree video, it has been demonstrated that viewport adaptive streaming systems[3][8][19][20], which use a tile-based approach, can reduce the bandwidth to deliver the video. The user's field of view (FoV) area can be extracted from the bitstream using a motion-constrained tile set (MCTS)[18]. The MCTS encoder limits the temporal inter prediction at tile boundaries, and tile can be extracted from a bitstream and transmitted to the client.

The aforementioned solutions were applicable for a single 360-degree video processing. However, using a single realistic 360-degree video, the user cannot move or walk in a VR environment. To overcome this problem, simultaneous transmission of multiple 360-degree videos is required to provide an immersive experience, that in turn, enables the user to move in a VR environment. As multiple video streaming requires multiple decoders at the client side, which is considerably challenging, the moving picture experts

Figure 1: Block diagram of viewport-dependent tiled 6DoF 360-degree video streaming

group (MPEG) has established the MPEG-Immersive (MPEG-I) sub-group to build a system that reduces the number of videos to be transmitted. MPEG-I defined three types of degrees of freedom (DoF) of users. First, three degrees of freedom (3DoF) supports only the rotations of the user's head. Second, three degrees of freedom plus (3DoF+) supports a limited movement of the user's head, while the user is sitting on a chair. Finally, six degrees of freedom (6DoF) supports full movement of the user including his or her entire body. Several studies implemented 3DoF+ systems by down-sampling multiple videos[10][11] and eliminating the correlation among[12]. In 2019, a test model for immersive video (TMIV) was proposed as a reference software for 6DoF video compression. It removes the correlation between multiple views to reduce the bandwidth, which is similar to multiview high-efficiency video coding (MV-HEVC)[9]. However, the aforementioned viewport-dependent streaming is not considered in TMIV.

Currently, TMIV supports viewport-independent compression of multiple 360-degree videos. Additionally, existing MCTS based streaming methods are designed for single 360-degree video. Therefore, this paper proposes a 6DoF viewport tile selector (VTS) and multiple-tile extractor to apply tiled streaming for multiple 360-degree videos. The contributions of this paper are two-fold: (i) Viewport area tiles selection for multiple 360-degree videos, which is compatible with TMIV and omnidirectional media format (OMAF)[4] and (ii) Viewport-dependent multiple-tile extractor. Figure 1 demonstrates the proposed streaming system. Input multiple 360-degree videos are pre-processed by the TMIV encoder; the server encodes the videos and generates two layers. The tile layer contains high-quality tiled bitstream; the viewport tiles are extracted from the tile layer using the VTS and the multiple-tile extractor. The base layer contains the low-quality bitstream that covers the entire area. The two layers are transmitted simultaneously to provide low-latency high-quality video. The client decodes the bitstreams, and the TMIV decoder generates the viewport which is displayed to the HMD.

The remainder of this paper is divided into the following sections. Section 2 provides the background. Section 3 describes the tiled 6DoF VTS and viewport-dependent multiple-tile extractor schemes. Experimental settings, results, and analyses are explained in Section 4. Finally, Section 5 provides conclusions and directions for future work.

Figure 2: Block diagram of TMIV architecture

2 BACKGROUND

3DoF+ and 6DoF provide an immersive experience in VR. However, they require the compression and streaming of multiple videos to support the user's body movement, which is considerably challenging with high-efficiency video coding (HEVC). As HEVC is designed for single video compression, it requires a large bandwidth and several decoders when handling multiple videos. To solve the aforementioned problems, MV-HEVC was proposed to compress multiview videos efficiently. MV-HEVC removes the correlation between multiple views at the codec level; in return, a MV-HEVC bitstream can be reconstructed by a MV-HEVC decoder, which implies that the existing hardware acceleration employed for HEVC cannot be used for MV-HEVC. Therefore, in MPEG-I, HEVC was employed as the reference software for 3DoF+. In January 2019, MPEG-I instituted a call for proposals (CfP) on 3DoF+[13]; its system architecture contains pre-processing and post-processing modules with the HEVC. Because the multiple-video correlation elimination is conducted at the system level, the system can apply future video codec, versatile video coding (VVC)[16]. In response to the CfP on 3DoF+, five proposals[5][7][15][21][22] were submitted in March 2019. Among these, the proposal of Technicolor and Intel[7] demonstrated the best results. Compared to HEVC anchor, the proposal demonstrates a bjontegaard delta rate (BD-rate) saving of 73.0% for the luma peak signal-to-noise ratio (PSNR) and an average pixel rate ratio saving of 73.34%.

Figure 3: TMIV based system architecture of viewport-dependent tiled 6DoF 360-degree video streaming

Based on the components of the proposed responses, MPEG-I established TMIV[17]. Notably, TMIV supports pre-processing and post-processing for streaming multiview videos to compress 6DoF videos more efficiently. The block diagram of TMIV is presented in Figure 2. As illustrated in the figure, TMIV eliminates the correlation among multiple videos. First, source views that are captured simultaneously from different positions are input to the view optimizer. The source view includes two kinds of videos: texture, and depth. The texture possesses the color information similar to a traditional 2D video, and the depth possesses the information about the distance between the camera and the objects on the texture. The view optimizer defines basic views and additional views. From the source views, the views that contain most of the objects in the scene are selected as the basic views. The remainder views are defined as additional views. Second, the basic and additional views are input to the atlas constructor, which includes the pruner, aggregator, and patch packer. The pruner removes the correlation between the basic views and the additional views. Further, the pruner removes the correlation among the additional views. Using depth, 2D textures are represented in the 3D space. If two points from different views have the same position on the 3D space, one point is removed from its view. Overlapping points are removed from the additional views, and the map of the removed area is defined as a mask. The mask represents valid and invalid pixels from the additional views. For example, in Figure 2, the black pixels represent the informative areas that have to be transmitted to the client side. The white pixels are the removed areas from the additional views because they

are already included in the basic views. Based on the mask, the patch packer extracts the valid pixels from the additional views and merges them to the occupancy map. The pixels are extracted as rectangles, and the rectangles are defined as a patch. However, if the objects move in the scene, the mask changes frame by frame. Accordingly, the size of the patch is different for each frame, and the locations of the patches are changed for each frame, which decreases the compression efficiency. Therefore, before the patch packer execution, the aggregator merges the masks over an intraperiod (e.g. 32 frames). The aggregator aggregates the masks, whereas the patch packer uses the same mask. Accordingly, in the intraperiod range, the extracted and merged patch locations are fixed, which increases the inter-frame compression efficiency. Once the occupancy map is generated, the patch packer extracts informative pixels from the additional view's texture, and depth; they are copied to the atlases. An atlas is a set of patches generated from multiple videos by TMIV. The number of videos to transmit decreases by generating atlases because only the residuals are extracted from the additional views and merged into the atlases. The basic views are completely copied into the basic views atlases. Consequently, the TMIV encoder generates the atlases and their metadata, and the video encoder (e.g. HEVC) encodes them to stream the videos to the client side. When the bitstreams and the metadata are transmitted to the client side, the TMIV decoder reconstructs the atlas patch occupancy map based on the metadata, which defines the informative areas of the atlases, and TMIV renderer generates a user's viewport based on the occupancy map and the atlases.

3 VIEWPORT-DEPENDENT 6DOF 360-DEGREE VIDEO TILED STREAMING

This section explains the VTS for 6DoF 360-degree video tiled streaming based on a user's head movement as well as the multiple-tile extractor. The proposed method applies viewport-dependent multiple 360-degree video tiled streaming on TMIV and HEVC, as depicted in Figure 3. The proposed system can provide high-quality 6DoF video with low delay and low bandwidth by simulcasting the complete low-quality 360-degree video and high-quality viewport tiles. A low-quality area is displayed for a short period even if the user turns his or her head, and the transmitted high-quality viewport tiles are decoded and rendered. The internal processes of the proposed system are as follows. Source views are pre-processed by the TMIV encoder, as illustrated in Figure 2. High-quality additional views atlases are encoded by the HEVC encoder without MCTS. The basic views atlases comprise two types of layers: tile layer and base layer. The base layer is encoded by the HEVC encoder without MCTS to provide an omnidirectional scene. By contrast, the tile layer, encoded by the HEVC encoder with MCTS, provides a high-quality video for the user's viewport. When the user's viewport information is received, the proposed VTS computes the user's viewport tiles on basic views atlases. The tiling for additional views atlases is not applied because additional views atlases contain the patches from different views. Then, viewport tiles on the basic views are extracted from the tile layer, and the extracted tiles from one basic view atlas are merged into one bitstream. The transmitted bitstreams are decoded, and the base layer and tile layer are merged. Finally, the TMIV decoder generates a viewport.

The remainder of this section includes the following of two subsections: (i) Viewport tile selection method for 6DoF 360-degree video that is compatible with OMAF and TMIV, which is explained in Section 3.1. (ii) Viewport-dependent multiple-tile extractor; this is introduced in Section 3.2.

3.1 Viewport Tile Selector for 6DoF 360-degree Videos

When a 360-degree video is transmitted and decoded at the client side, a renderer generates a viewport depending on the user's head movement. To use the 2D video codec, points on the 3D sphere are projected onto a 2D plane video. Equirectangular projection (ERP) is one of the projection methods, which is applied in this paper. This section first introduces a method for the viewport tile selection on a single 360-degree ERP video based on Yu et al[24], and multiple 360-degree videos tiles selection method based on TMIV. In OMAF[4] and TMIV[17], the user's head rotation is represented by the Euler angle. The Euler angle includes yaw, pitch, and roll, and they correspond to rotation about the Z, Y, and X axes, respectively. Each angle is represented using degrees, which is converted into radians to compute the viewport area, as shown in Equation 1, where α , β , and γ stands for yaw, pitch, and roll, respectively.

$$\begin{aligned}\alpha &= \alpha_d \cdot \pi / 180 \\ \beta &= \beta_d \cdot \pi / 180 \\ \gamma &= \gamma_d \cdot \pi / 180\end{aligned}\quad (1)$$

Matrix R represents the rotation matrix that reflects the user's head movement. As denoted in Equation 2, matrix R comprises the product of the rotation matrices about the X, Y, and Z axes.

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Matrix K contains information about a viewport camera intrinsic matrix, as shown in Equation 3. Here, f_x and f_y denote the focal lengths of the camera. Let fov_x denote the horizontal FoV represented using radian, and W_v denote the width of the viewport. f_x can be computed as $f_x = (W_v/2) \cdot (1/\tan(fov_x/2))$. c_x and c_y denote the coordinates of the principal point C in the viewport.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Let $p = [u, v, 1]^T$ denote the 2D homogeneous coordinates of the viewport. Let P denote the viewport points on the 360-degree video, represented by Cartesian coordinates that can be computed using Equation 4. In the equation, the product of R, inverse matrix of K, and point p is divided by the L2 norm of $K^{-1} \cdot p$.

$$P = R \cdot \frac{K^{-1}p}{\|K^{-1}p\|_2} \quad (4)$$

To obtain the coordinates of the 2D 360-degree ERP video, points in spherical coordinates are required. A point, P , in Cartesian coordinates can be converted to a point in spherical coordinates using Equation 5.

$$\begin{aligned}\phi &= \text{atan2}(P_y, P_x) \cdot 180/\pi \\ \theta &= \text{asin}(P_z) \cdot 180/\pi\end{aligned}\quad (5)$$

The computed spherical coordinates can be converted to a point of the 2D 360-degree ERP video using Equation 6.

$$\begin{aligned}x &= \text{width} \cdot (0.5 + \phi/360) \\ y &= \text{height} \cdot (0.5 + \theta/180)\end{aligned}\quad (6)$$

If the point p is computed using Equation 6, a tile index that contains p can be obtained by Equation 7, where t_i , pw , ph , tw , and th represent tile index, picture width, picture height, tile width, and tile height, respectively.

$$t_i = (y/th) \cdot (pw/tw) + x/tw \quad (7)$$

The aforementioned method is used to select viewport tiles of a single 360-degree video, that employs OMAF and TMIV. However, for 3DoF+ and 6DoF, viewport tiles selection for multiple 360-degree videos is required. Therefore, we have designed a warping based multiple 360-degree videos viewport tiles selection method. In the traditional single 360-degree video viewport tiles selection method, the position of the user and the video are the same. However, in 6DoF, these locations are not always the same. This makes 6DoF viewport tiled streaming comparatively more challenging. To detect the user's viewport area from each source view, several points of

---

Algorithm 1: Multiple 360-degree videos heuristic view-
port tile decision

PW : Picture width
 PH : Picture height
 TW : Tile width
 TH : Tile height
 VW : Viewport width
 VH : Viewport height
 H_i : Heuristic interval coefficient
 S : Source views
 V : Viewport
 p_s : Points of a source view
 p_v : Points of a viewport
 t_p : Tile index of a point p
 L_t : List of viewport tiles in a source view
for all $S_i \in S$ **do**
  for $row = 0$ to $row = PW/TW$ **do**
    for $col = 0$ to $col = PH/TH$ **do**
      for $i = 0$ to $i = H_i$ **do**
        add $S_i(row \cdot TW + i, col \cdot TH + i)$ to p_s
      end for
    end for
  end for
  for $row = 0$ to $row = VW/TW$ **do**
    for $col = 0$ to $col = VH/TH$ **do**
      for $i = 0$ to $i = H_i$ **do**
        add $V(row \cdot TW + i, col \cdot TH + i)$ to p_v
      end for
    end for
  end for
  for all $S_i \in S$ **do**
     $R \leftarrow$ Rotation matrix from S_i to V
     $T \leftarrow$ Translation matrix from S_i to V
    for all $p_s \in S_i$ **do**
       $t_{p_s} = (p_s(y)/TH) \cdot (PW/TW) + p_s(x)/TW$
       $p'_s = R \cdot p_s + T$
      if t_{p_s} is not in L_t **then**
        for all p_v **do**
          if $p'_s = p_v$ **then**
            add t_{p_s} to L_t
          end if
        end for
      end if
    end for
  end for
end for

---

the source views are projected into the 3D space. When a viewport at a user's position is decided, several points of each source view are projected into 3D space and reprojected into the viewport position to detect the viewport tiles. This process is called warping. The projection from a viewport to each source view is not conducted because warping requires a source view depth map. However, the server does not always have the viewport position depth map because the user can move toward any direction in 6DoF. Therefore,

Figure 4: Functional flow chart of the proposed multiple-tile extractor

projection from each source view to a viewport is processed to determine the viewport tiles.

Warping includes three steps. First, the input image coordinates are unprojected into the world coordinates using the camera parameter and depth map. The depth map provides the distance between a camera and the object displayed in the image. During unprojection, the input image is projected onto the world coordinates. Second, an affine transformation is applied to each point in the world coordinates. As shown in Equation 8, the input image coordinates are rotated using rotation matrix R and moved by translation matrix t . In this step, the input image coordinates are transformed into the target image coordinates. Finally, the transformed world coordinates are projected onto the target image.

$$x' = Rx + t \quad (8)$$

As explained before, warping is conducted from each source view to the viewport to determine the viewport tiles. Algorithm 1 describes the multiple 360-degree videos viewport tile selection, based on TMIV, in detail. To implement a heuristic tile selection algorithm for achieving low-delay streaming, a part of the source view points and viewport points are added to p_s and p_v . Heuristic interval coefficient H_i is used to insert the points at regular intervals. In our experiment, H_i was set to 48. Rotation matrix R and

translation matrix T from i th source view S_i to V is computed. For each point p_s of S_i , a single affine transformation using matrix R and T is conducted. The warped point p'_s denotes the transformed point from S_i to V . t_{p_s} denotes the tile index which includes point p'_s . When point p'_s matches one of the points p_v , t_{p_s} belongs to the viewport tile, and it is added into L_t . If the tile index t_{p_s} is already included in the tile list L_t , p'_s is not compared with p_v to reduce the computational complexity. Else, if the tile index t_{p_s} is not in tile list L_t , point p'_s is compared with every p_v . If both the points are in the same position, it indicates that point p_s belongs to the viewport area, and t_{p_s} should be added into L_t .

3.2 Viewport-dependent Multiple-tile Extractor

Viewport tiles selection method was introduced in Section 3.1, and the selected tiles are extracted from the basic views bitstreams as illustrated in Figure 3. The HEVC test model (HM) 16.20 includes a MCTS tile extractor. However, it can extract only one tile and generate a bitstream containing only one tile. It generates a number of tiles bitstreams, and it requires a number of decoders at the client side to implement low delay rendering. To solve this problem, this paper introduces an implementation of a multiple-tile extractor which extracts multiple tiles and generates one bitstream.

Figure 4 presents the functional flow chart of the proposed extractor. An HEVC bitstream includes several types of network abstraction layer (NAL) units. The video parameter set (VPS), sequence parameter set (SPS), and picture parameter set (PPS) contain the bitstream information. A slice contains a header and a compressed video data field. The proposed extractor decodes the input bitstream, finds PPS, and parses the required information, such as the number of tiles, picture size, tile size array, and coding tree unit (CTU) size. In a MCTS bitstream, the parameter sets of each tile are saved as extraction information sets (EIS) supplemental enhancement information (SEI) messages. This is because, in a HEVC decoder, the parameter sets are stored in the decoder instance, and the extractor cannot extract all of the parameter sets from the decoder. Hence, the EIS SEI messages, containing the parameter sets of tiles, are stored in the MCTS bitstream. Therefore, after parsing the PPS of the input bitstream, the extractor parses the EIS SEI messages to obtain the parameter sets. Then, the extractor identifies the slices that contain the tiles to be extracted. In MCTS based tiled encoding, one slice contains one tile. Slices have segment addresses; thereby the extractor can identify the target tile by comparing the input slice's segment address and the target slice's segment address. Therefore, the extractor identifies and extracts the target slices using the parsed information (such as the tile size array and CTU size). Meanwhile, the parameter sets obtained from the EIS SEI messages are for a single tile bitstream. However, the proposed extractor generates the multiple tiles bitstream that has the same size as that of the input bitstream. Therefore, after extracting the target slices, the extractor replaces certain options from the parameter sets such as picture size, loop filter options, and tiles enabled flag. Then, the extractor encodes the parameter sets and slice header and converts the input slices NAL units to the output slices NAL units. Finally, the output bitstream containing multiple tiles is generated.

Table 1: Experiment settings

Items	Experimental values
Texture QP	22, 27, 32, 37, 42
Depth QP	9, 14, 21, 30, 38
GOP, framerate	GOP: 16, framerate: 30 fps
Intraperiod	32
FoV	$90^\circ \times 90^\circ$
Evaluation frames	97 frames
Segment duration	32, 32, 33 frames

Table 2: Characteristics of the immersive video test sequence

Sequence	Class	Resolution	No. of source views	No. of basic views
<i>ClassroomVideo</i>	CG1-A	4096×2048	15	1
<i>TechnicolorMuseum</i>	CG1-B	2048×2048	24	2
<i>TechnicolorHijack</i>	CG1-C	4096×4096	10	1

The proposed extractor has advantages over the existing single tile extractor. It can reduce the number of required decoders as well as the decoding time. The detailed experimental results of the proposed extractor will be discussed in Section 4.2.

4 EXPERIMENTAL RESULTS

4.1 Experimental Conditions

This section introduces the experimental conditions of the proposed 6DoF streaming system. One server was used for this experiment, with two Intel Xeon E5-2687w v4 CPUs (24 cores, 48 threads), 128 GB of memory, GTX 1080 Ti, and Ubuntu 18.04 was installed. For the client, a Windows 10 desktop with Intel i7-7700k CPU (4 cores, 8 threads), 16GB of memory, and GTX 1080 Ti was used. This experiment was conducted in the common test conditions (CTC) for immersive video[14] defined by MPEG-I. The CTC defines the test options such as the test sequence, encoding options, reference softwares, and the quality assessment metric for a fair evaluation. TMIV version 3.0 and HM 16.20 were used in this experiment. Pre-processing and post-processing of multiple 360-degree videos were conducted by TMIV, and the pre-processed videos were encoded by HM. The experimental parameters are presented in Table 1. According to the CTC, the test materials were encoded using five quantization parameters (QP). Different QP values were applied to both texture and depth. The first four QP values were used to encode the tile layer (high quality); the last QP value was used to encode the base layer (low quality). The group of pictures (GOP) value was set to 16, and the framerate was 30 fps. The FoV of the viewport was $90^\circ \times 90^\circ$, and the evaluation frames were 97 frames. Considering the streaming scenario, the videos were divided into chunks of 32, 32, and 33 frames. The sum of viewport tiles of the frames in a chunk is selected. The viewport tiles were computed using the proposed VTS, and they were extracted from the tile layer basic views using the proposed multiple-tile extractor.

Table 3: Average number of viewport tiles per frame for each movement scenarios

Class	Tile size	No. of tiles	Movement scenario				No. of tiles savings
			Case 1	Case 2	Case 3	Average	
CG1-A	512×512	32	8.00	9.20	10.45	9.22	71.18%
CG1-B		16	5.00	4.50	5.26	4.92	69.25%
CG1-C		64	23.00	20.80	25.06	22.95	64.14%
Average	512×512	37.33	12.00	11.50	13.59	12.36	68.19%

Table 4: Tiled streaming performance of different tile sizes

Class	BD-rate		
	Y-PSNR		
	256×256	512×512	1024×1024
CG1-A	−16.35%	−19.50%	−11.28%
CG1-B	−24.95%	−21.86%	−11.20%
CG1-C	10.07%	5.24%	4.37%
Average	−10.41%	−12.04%	−6.03%

Meanwhile, at the client side, FFmpeg with hevc_nvenc (NVIDIA NVENC HEVC encoder) was used to measure the decoding performance using GPU acceleration. After decoding, TMIV decoder generated the user’s viewport using the reconstructed videos. The user’s movement data were needed to generate the viewport. We used the user’s movement scenarios defined by MPEG-I CTC, called pose traces. There were three pose traces for each test sequence, and they contained the user’s head movements as well as the body movements.

For the quality assessment, the following metrics were used as defined in CTC: PSNR, video multimethod assessment fusion (VMAF)[1], multi-scale structural similarity (MS-SSIM)[23], and immersive video PSNR (IV-PSNR)[6]. Simultaneously captured multiple 360-degree videos could be used as test materials. However, to compress the test materials using TMIV, the camera parameters should be provided, which contains the location and translation of each camera. Further, as mentioned in Section 2, multiview videos have texture and depth, and each component represents color and distance information. Therefore, we used test sequences defined in MPEG-I CTC to meet the requirements. MPEG-I provided eight immersive video test sequences that contain texture, depth, and camera parameters. In this experiment, three 360-degree video test sequences are used, as shown in Table 2. The test sequences, *ClassroomVideo*, *TechnicolorMuseum*, and *TechnicolorHijack* are classified by their class name, CG1-A, CG1-B, and CG1-C, respectively. The basic views from the test sequences were computed by the view optimizer of TMIV, as explained in Section 2.

4.2 Result Analysis

This section introduces and analyzes the experimental results of the proposed 6DoF streaming system. In tiled streaming, the tile size is vital, as described by Zare et al.[25]. As mentioned in Section 4.1, we selected three test sequences for each test sequence to determine the best tile size. The characteristics of these test sequences are as follows. CG1-A contained several objects and a very complex background. On the contrary, CG1-C contained fewer objects than

Figure 5: RD-curve, TMIV anchor vs 256×256, 512×512, and 1024×1024 tiled streaming

CG1-A and had a simple background. Meanwhile, CG1-B contained fewer objects and a less complex background than CG1-A, but it contained more objects and a more complex background than CG1-C. We encoded these sequences using MCTS with 256×256, 512×512, and 1024×1024 tile sizes. In traditional 3DoF 360-degree video tiled streaming, a smaller tile encoding was efficient compared to a bigger tile encoding because fewer areas were transmitted to the client side. However, in 6DoF 360-degree video tiled streaming, there was a trade-off between different tile sizes. Table 4 displays the tiled streaming performance for different tile sizes. Among them, a tile size of 512×512 provided the best BD-rate. Even though a tile size of 256×256 contained the smallest areas of tiles, it did not provided the best result. If there are several tiles, the corresponding number of slice headers must be included. Further, dense tiling decreased the objective and subjective quality of the decoded video because MCTS disabled the loop filter at the edges of the tiles. In CG1-A, the bit-rate of the tiles decreased with decrease in the tile size. However, the objective quality of the generated viewport decreased with a decrease in the tile size because of the quality degradation of the decoded video. In CG1-C, most of the objects were already included in the viewport tiles, and the background was simple. Therefore, the BD-rate of a bigger tile size showed the best result. In CG1-B, a smaller tile size demonstrated the best result, similar to that demonstrated by the traditional 3DoF 360-degree video tiled streaming.

Figure 6: Performance comparison of the single and multiple-tile extractors regarding (a) decoding memory use and (b) decoding time

As depicted in Figure 5, tiled streaming provided better results compared those obtained via non-tiled streaming. Among the different tile sizes, a tile size of 512×512 provided the best results, and we decided to encode the basic views of the tile layer with a tile size of 512×512.

To evaluate the performance of the proposed multiple-tile extractor compared to that of the existing single tile extractor, we also conducted an experiment to measure the decoding memory use and decoding time consumption. The aforementioned three test sequences and three tile sizes were used in this experiment. As explained in Section 4.1, we used FFmpeg hevc_nvenc decoder. Figure 6 illustrates the experimental results of the single tile extractor and the proposed multiple-tile extractor. The single tile extractor consumed an average of 82.94 GB of memory and 75.73 s for decoding the basic view texture and depth tile bitstreams, whereas the proposed multiple-tile extractor consumed an average of 33.09 GB of memory and 33.69 s. Thus, the advantage of using the proposed multiple-tile extractor is verified.

Table 3 depicts the number of viewport tiles for each test sequence and movement scenarios. Only 31.81% of high-quality tiles of the tile layer were transmitted, and the base layer included the entire low-quality 360-degree video. Table 5 presents the BD-rate savings of the proposed tiled streaming method compared to the TMIV anchor. The proposed method demonstrated overall better results compared to those of traditional non-tiled streaming. With regard to luma PSNR, VMAR, MS-SSIM and IV-PSNR, the proposed method provided an average BD-rate savings of 12.04%, 14.52%, 12.02%, and 15.28%, respectively. According to the input documents of MPEG-I[6], IV-PSNR showed a correlation coefficient of 0.7177 with the subjective score (mean opinion score, MOS), which is the highest among the quality metrics considered in MPEG-I. Meanwhile, PSNR showed a correlation coefficient of 0.2109, which is the lowest. Therefore, the proposed method is advantageous based on human’s quality assessment because it showed better results on IV-PSNR than luma-PSNR.

5 CONCLUSION

This paper proposed a viewport-dependent tiled streaming system for 6DoF 360-degree video. The system included a viewport tile

Table 5: BD-rate savings of the proposed tiled streaming method compared to the TMIV anchor

Class	BD-rate			
	Y-PSNR	VMAF	MS-SSIM	IV-PSNR
CG1-A	−19.50%	−27.82%	−21.92%	−30.02%
CG1-B	−21.86%	−21.75%	−20.82%	−21.06%
CG1-C	5.24%	5.99%	6.68%	5.23%
Average	−12.04%	−14.52%	−12.02%	−15.28%

selector and multiple-tile extractor to reduce the bit-rate and decoding resource usage. More specifically, when a viewport area on a single video at a user’s position was decided, the viewport areas on multiple 360-degree videos were selected by the warping from the multiple 360-degree videos to the viewport position. The selected tiles were extracted from the source views as one bitstream and transmitted to the client. The proposed method showed a 12.04% BD-rate saving for the luma PSNR compared to those obtained via the TMIV anchor and a 55.51% decoding time saving compared to those obtained via the existing tiled streaming that uses a single tile extractor. Moreover, the proposed system had advantages over the MV-HEVC based streaming system because the proposed system could apply any video codec, such as advanced video coding (AVC), HEVC, and VVC. In future, we intend to conduct experiments to obtain an equation that determines a compromise point between bit-rate saving and tile size. Further, we plan to implement a real-time streaming system that supports 6DoF VR using parallel decoding and optimization.

ACKNOWLEDGMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2018-0-00765, Development of Compression and Transmission Technologies for Ultra High-Quality Immersive Videos Supporting 6DoF). This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00231, Development of Low Latency VR·AR Streaming Technology based on 5G edge cloud).

REFERENCES

- [1] Christos G Bampis, Alan C Bovik, and Zhi Li. 2018. A Simple Prediction Fusion Improves Data-driven Full-Reference Video Quality Assessment Models. In *2018 Picture Coding Symposium (PCS)*. IEEE, 298–302.
- [2] Mary-Luc Champel, Thomas Stockhammer, Thierry Fautier, Emmanuel Thomas, and Rob Koenen. 2016. Quality Requirements for VR. 116th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG116/m39532.
- [3] Y Sánchez de la Fuente, Robert Skupin, and Thomas Schierl. 2017. Video processing for panoramic streaming using HEVC and its scalable extensions. *Multimedia Tools and Applications* 76, 4 (2017), 5631–5659.
- [4] Sachin Deshpande, Ye-Kui Wang, and Miska M. Hannuksela. 2019. Preliminary WD of ISO/IEC 23090-2 2nd edition OMAF. 127th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG127/n18587.
- [5] Marek Domanski, Adrian Dziembowski, Dawid Mieloch, Olgierd Stankiewicz, Jakub Stankowski, Adam Grzelka, Gwangsoon Lee, and Jeongil Seo. 2019. 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). 126th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG2019/m47407.
- [6] Adrian Dziembowski. 2019. Software manual of IV-PSNR for Immersive Video. 128th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG127/n18709.

- [7] Julien Fleureau, Franck Thudor, Renaud Dore, Basel Salahieh, Maxym Dmytrychenko, and Jill Boyce. 2019. Technicolor-Intel Response to 3DoF+ CFP. 126th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG2019/m47445.
- [8] Ramin Ghaznavi-Youvalari, Miska M Hannuksela, Alireza Aminlou, and Moncef Gabbouj. 2017. Viewport-dependent delivery schemes for stereoscopic panoramic video. In *2017 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. IEEE, 1–4.
- [9] Miska M Hannuksela, Ye Yan, Xuehui Huang, and Houqiang Li. 2015. Overview of the multiview high efficiency video coding (MV-HEVC) standard. In *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2154–2158.
- [10] JongBeom Jeong, Dongmin Jang, Jangwoo Son, and Eun-Seok Ryu. 2018. 3DoF+ 360 Video Location-Based Asymmetric Down-Sampling for View Synthesis to Immersive VR Video Streaming. *Sensors* 18, 9 (2018), 3148.
- [11] JongBeom Jeong, Dongmin Jang, Jang-Woo Son, and Eun-Seok Ryu. 2018. Bitrate efficient 3DoF+ 360 video view synthesis for immersive VR video streaming. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 581–586.
- [12] Jong-Beom Jeong, Soonbin Lee, Dongmin Jang, and Eun-Seok Ryu. 2019. Towards 3DoF+ 360 Video Streaming System for Immersive Media. *IEEE Access* 7 (2019), 136399–136408.
- [13] ISO/IEC JTC1/SC29/WG11. 2019. Call for Proposals on 3DoF+ Visual. 125th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG/n18145.
- [14] Joel Jung, Bart Kroon, and Jill Boyce. 2019. Common Test Conditions for Immersive Video. 128th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG128/n18789.
- [15] Bart Kroon and Bart Sonneveldt. 2019. Philips response to 3DoF+ Visual CFP. 126th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG2019/m47179.
- [16] Jens-Rainer Ohm and Gary J Sullivan. 2018. Versatile Video Coding—towards the next generation of video compression. In *Picture Coding Symposium 2018*.
- [17] Basel Salahieh, Bart Kroon, Joel Jung, and Marek Domański. 2019. Test Model 3 for Immersive Video. 128th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG128/n18795.
- [18] Robert Skupin, Yago Sanchez, Karsten Sühling, Thomas Schierl, Eun-Seok Ryu, and Jangwoo Son. 2018. Temporal MCTS Coding Constraints Implementation. 122th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG 122/m42423.
- [19] Jangwoo Son, Dongmin Jang, and Eun-Seok Ryu. 2018. Implementing 360 video tiled streaming system. In *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, 521–524.
- [20] Jangwoo Son and Eun-Seok Ryu. 2018. Tile-based 360-degree video streaming for mobile virtual reality in cyber physical system. *Computers & Electrical Engineering* 72 (2018), 361–368.
- [21] Vinod Kumar Malamal Vadakital, Kimmo Roimela, Lauri Ilola, Jaakko Keränen, Mika Pesonen, Sebastian Schwarz, Jani Iainema, and Miska Hannuksela. 2019. Description of Nokia's response to CFP for 3DOF+ visual. 126th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG2019/m47372.
- [22] Bin Wang, Yule Sun, and Lu Yu. 2019. Description of Zhejiang University's response to 3DoF+ Visual CFP. 126th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG2019/m47684.
- [23] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. Ieee, 1398–1402.
- [24] Matt Yu, Haricharan Lakshman, and Bernd Girod. 2015. A framework to evaluate omnidirectional video coding schemes. In *2015 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 31–36.
- [25] Alireza Zare, Alireza Aminlou, Miska M Hannuksela, and Moncef Gabbouj. 2016. HEVC-compliant tile-based streaming of panoramic video for virtual reality applications. In *Proceedings of the 24th ACM international conference on Multimedia*. 601–605.