A Proposal for Zoom-in/out View Streaming based on Object Information of Free Viewpoint Video

Minjae Seo[1], Jong-Ho Paik[1] and Gooman Park[2*]
[1]Department of Software Convergence, Seoul Women's University, Seoul, South Korea
[2]Department of Electronic and IT Media Engineering, Seoul National University of Science Technology, Seoul, South Korea
[e-mail: seominjae@swu.ac.kr, paikjh@swu.ac.kr, gmpark@seoultech.ac.kr]
*Corresponding author: Gooman Park

Received September 24, 2021; revised February 2, 2022; accepted March 4, 2022; published March 31, 2022

Abstract

Free viewpoint video (FVV) service is an immersive media service that allows a user to watch it from a desired location or viewpoint. It is composed of various forms according to the direction of the viewpoint of the provided video, and includes zoom in/out in the service. As consumers' demand for active watching is increasing, the importance of FVV services is expected to grow gradually. However, additional considerations are needed to seamlessly stream FVV service. FVV includes a plurality of videos, video changes may occur frequently due to movement of the viewpoint. Frequent occurrence of video switching or re-request another video can cause service delay and it also can lower user's quality of service (QoS). In this case, we assumed that if a video showing an object that the user wants to watch is selected and provided, it is highly likely to meet the needs of the viewer. In particular, it is important to provide an object-oriented FVV service when zooming in. When video zooming in in the usual way, it cannot be guaranteed to zoom in around the object. Zoom function does not consider about video viewing. It only considers the viewing screen size and it crop the video view as fixed screen location. To solve this problem, we propose a zoom in/out method of object-centered dynamic adaptive streaming of FVV in this paper. Through the method proposed in this paper, users can enjoy the optimal video service because they are provided with the desired object-based video.

Keywords: Free viewpoint video, FVV service, video streaming service, video zoom in/out

A preliminary version of this paper was presented at APIC-IST 2021, and was selected as an outstanding paper. This study was supported by the Research Program funded by the SeoulTech(Seoul National University of Science and Technology).

http://doi.org/10.3837/tiis.2022.03.010 **ISSN : 1976-7277**

1. Introduction

Due to the advancement of image processing technology and the development of network technology, high-definition video recording equipment has become common, and various video services are being distributed on various platforms [1]. Recently, research to create a new media service is being actively conducted centered on terminal manufacturers and telecommunication providers. As one of them, studies for providing a Free-viewpoint video (FVV) service are being actively conducted. FVV service is an immersive media service that composes and provides videos from various viewpoints in one content, allowing users to enjoy them at a desired location or viewpoint [2]. It is composed of various forms according to the direction of the viewpoint of the provided video, and even the adjustment of the viewing depth such as zoom in/out function is included in the service [3].

These characteristics of FVV services are consistent with recent media consumption patterns. For consumers who want to watch a specific person in detail even on a single stage, videos called “Fancam” are created and distributed from different viewpoints through separate capturing or editing. In other words, it proves that consumers' demand for active viewing is increasing and more and more people want to watch only a specific part of the stage. The various viewpoint directions and zoom in/out functions provided by FVV service can reflect these needs, so the importance of FVV service is expected to grow gradually.

However, additional considerations are needed to stream stably FVV to users. Since FVV includes a plurality of videos, and view changes may occur frequently due to movement of the viewpoint. It is necessary to provide a plurality of videos while guaranteeing a user's desired view, but it is physically impossible to provide FVV service as streaming at a time because of the massive amount of video constituting one FVV content. At this time, it can be an important point to maintain content quality of service (QoS) to prevent service delay due to frequent video switching or re-request by selecting and providing other videos so that the user can watch the desired video.

Considering these problems, video selection based on specific part of FVV contents may increase the relevance between selected videos. A specific part that the user wants to watch may correspond to constituting the screen which likely to be a person or object. If a video view in which a specific object appears well is selected and provided, it is highly likely to meet the video needs of viewers who want to view the object. This may reduce the number of times requesting video while not reducing the service quality for the user. In particular, it is important to provide an object-oriented FVV service in case of zooming in. When video zooming-in in the usual way, it cannot be guaranteed to zoom in the view around the object. Although the zoom in/out function is included in the definition of FVV service, the details of how to apply the function are rarely considered. However, in order to satisfy the quality of experience (QoE) of users who want more realistic videos and interactive viewing than existing ways for FVV service, it is important to bind the zoom-in method and the target.

To solve this problem, we propose a zoom in/out method of object-centered dynamic adaptive streaming of FVV in this paper. In the proposed method, the optimal recognition view for each object is searched among videos from multiple viewpoints, and the information of the object is extracted to construct the video playback information.

In order to maximize the video view based on the object of video service, the video playback time information was considered in which the object is optimally recognized. For the FVV service that moves viewpoint video frequently, we propose a method of selecting and requesting videos based on object information, and providing coordinate information so that zoom in/out function can be worked based on the provided video. By zooming in/out through the provided coordinate information, the user can maintain object-centered viewing stably. Through the provided information, the client selects and requests FVV service to ensure the user's object-centric zoom in/out video. Through the method proposed in this paper, users can enjoy the optimal video service, because they are provided with the desired object-based video.

2. Related Works

2.1 Free Viewpoint Video (FVV)

FVV service is a video technology that allows users to enjoy the content by adjusting the desired position and angle at any point in the content [4]. By providing video from multiple viewpoints included in one content at the same time to support more realistic and active viewing than the existing fixed-view video service, users can change the video on the screen by themselves to move the viewpoints and angles [5]. **Fig. 1** shows an example of the configuration of FVV service. It mainly targets a stage such as a stadium, and captures and acquires videos by setting cameras at various locations and spots. The number of videos may be tens or more, and the captured images are stored in FVV content server to generate and manage FVV contents services.

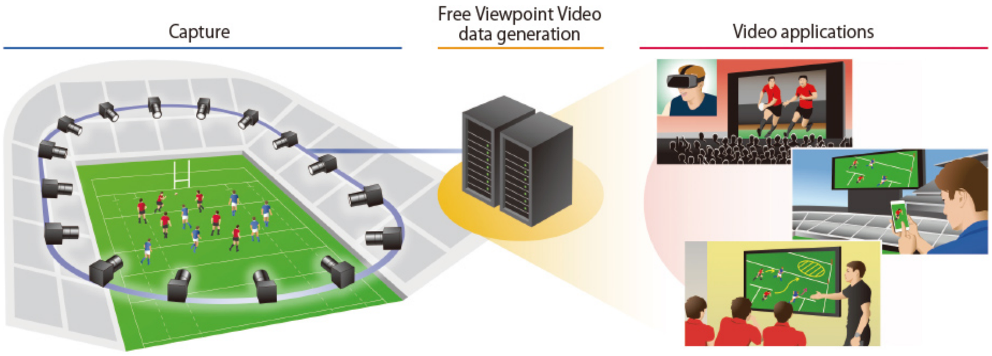


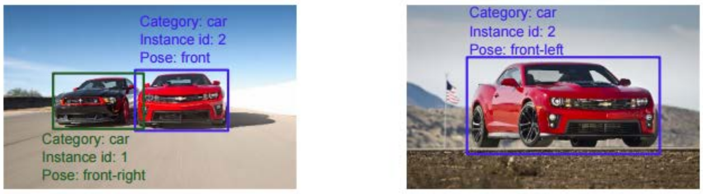
Fig. 1. Configuration example of Free-viewpoint video service [6]

One of the important features of FVV is that metadata related to a plurality of videos exists in various forms. FVV service is characterized in that it does its best to provide the maximum possible when a user wants to watch a location and space where video does not exist between each captured view, not just simply providing multiple views. Through video stitching or video synthesis for non-existent video angles, it aims to provide users with a more active viewing experience than existing video services. In the case of a large-capacity video service including a lot of data, such as FVV service, the processing time of client requests and video search speed can determine quality of service (QoS). Therefore, in order to find the requested video quickly in response to the user's selection and change, video storage rules and efficient request response and processing methods are required. This rule is not sufficient to consider only the storage method of the video, and it is necessary to determine it in connection with the transmission method. Therefore, it is important to deal with video storage method and transmission method for transmission.

In immersive media part MPEG-I in Moving pictures expert group (MPEG) standard, where FVV research is being conducted, 6 Degrees of freedom (DoF) movement, a similar immersive media, has recently been limited to movement within a few steps from free movement [7]. It means that there are practical restrictions in providing free movement, and it is not difficult to think that these restrictions are also applied to FVV which closely related with it. However, FVV is a service that not only provides captured videos, but also provides an intermediate view between two video views through synthesis when there is a location that the user wants to watch between the videos. Therefore, for FVV service including providing a synthesis view, it is inevitably limited to a rather restrictive movement line. However, this paper does not deal with the viewpoint synthesis technology, and we studied for focusing on the range of zoom in/out in FVV services.

2.2 Object Co-detection

In this paper, object co-detection technique is referred to apply the similarity between each video and object-based information within the free-viewpoint taken from multiple viewpoints. Object joint detection technology recognizes an object based on images taken from multiple angles of the same object and then determines whether it is the same object. This information is a technology for recognizing that an object within a separate video view is the same object, and not only improves the object identification accuracy during video movement, but also enables detection-based movement direction tracking [8]. Fig. 2 shows two images that appear the same red car. Even though (a) and (b) image is captured in different angle and background, but it is noticed as same one.



(a) Input Image #1 (b) Input Image #2

Fig. 2. Example of object co-detection [8]

Table 1. Average precision rate of object co-detection

Average Precision (%)		car	Pedestrian
Stereo Pair	prior method	49.8	59.7
	co-detector	53.5	62.7
Random Pair	prior method	49.8	59.7
	co-detector	50.0	58.1

In the case of a person, information about a specific person can be co-detected and mapped to a classification value. However, it is not only important to check whether a person is located, but a reference value is needed to determine whether an object can be seen "well" when the recognition rate of the object is high. Table 2 shows the average precision rate of object co-detection. It was applied to object precision by referring to the corresponding the detecting value. There is similar study about the multi tracking with random cameras for pedestrians, and it is applicable for this paper if the detecting range can be extended [9]. However, since there are practical limitations to directly applying the object co-detection technology, this paper applied AlphaPose and Face Recognition, which are public object recognition technologies for simulation, based on this basis [10].

2.3 Dynamic Adaptive Streaming over HTTP (DASH)

In this paper, it is assumed that DASH is used for streaming service. DASH is an international standard for http-based adaptive streaming and is widely used in video services [11]. DASH pursues with the goal of providing a seamless media service by providing it according to the user's environment such as network status and terminal performance [12]. DASH includes 5 technical formats within the scope of the standard. In this paper, we focus on media presentation document (MPD) description part for content consumption and the media segment part for media time-division in order to provide FVV service [13].

MPD shown in **Fig. 3** is a manifest document defined for media content consumption in DASH. MPD includes URL addresses of all playable content related to the media content service, as well as the classification of alternative media that can be selected according to the user's environment. MPD contains accessible and usable segments and their corresponding time information. In DASH-based media streaming service, three types of time related information may be considered. First one is the time expressed by *Period*. *Period* means a playback time within a section divided into arbitrary times for media service and has a continuous form [14]. *Representation* within a *Period* share a common representation reproduction time. In addition, *Representation* of an Access Unit (AU), which is a media unit that can be accessed and played, in the media stream also shares a common playback time. In order to synchronize different media components, such as video or audio, or to replace the same media with different encoded ones, it is important to share the same timeline. The last time information is segment *Availability*.

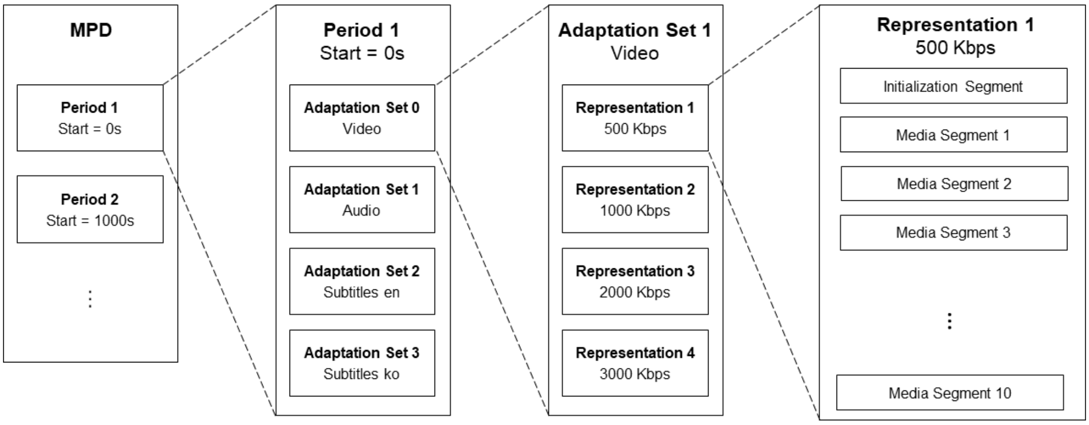


Fig. 3. Structure of MPD

In *AdaptationSet*, one or more media data such as video, audio, subtitles, etc. are each composed, and the same media is encoded and included in different ways so that it can be replaced. *Representation* is a structure that describes so that each media data encoded differently in bit rate, resolution, codec, etc. can be classified and selected in *AdaptationSet*, and includes one or more media elements. For example, in the case of a video, even videos reproduced in the same time period may be provided in different sizes depending on the user's environment. That is, when various environmental changes such as codec, language, and resolution occur in the client, unlike the condition of the initially selected data, redundant information is included so that the service can be guaranteed through the replaceable media in *AdaptationSet* and *Representation*. In this paper, we considered a method of composing a video in content using the corresponding information.

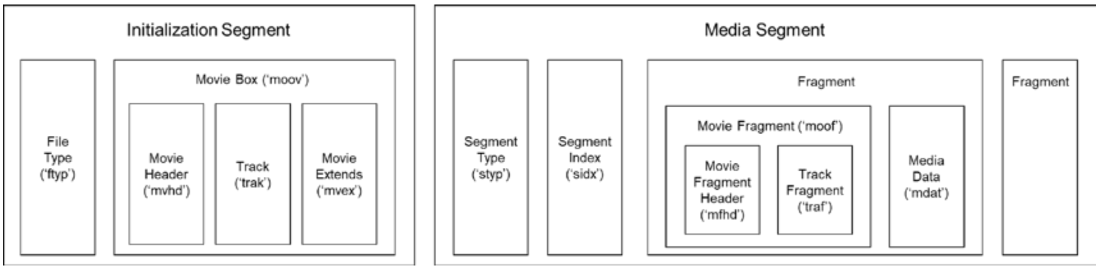


Fig. 4. Structure of ISOBMFF for streaming service

The media segment divided for data transmission of content includes the divided media stream required for actual playback. The media segment of DASH handles files in ISOBMFF format. ISOBMFF file defines the structure and data inclusion method for the expression of timed media data such as audio and video, including non-timed media information such as metadata and etc. [15].

As shown in **Fig. 4**, ISOBMFF organizes information in the form of boxes. A box is a byte type stream, and all data of ISOBMFF is included in the box. ISOBMFF exists in a form for a service for storage, but also in a segmented form for streaming media such as DASH. The box type has a highly extensible structure and can be entered as an option except for the main essential boxes, and the position of the additionally input box is not subject to great restrictions and is compatible with existing boxes. **Fig. 4** shows the basic ISOBMFF structure for media streaming. The initialization segment needs to be acquired only once initially, but the media segment is provided in the form of a plurality of segments according to the length. In this paper, we consider a method of providing object information using the extensibility of ISOBMFF. We would like to propose an additional box type so that object information can be updated when every media segment is acquired.

There are many related works that had studied about FVV streaming over DASH. One of them studied about a DASH-based free-viewpoint video streaming system, and it tried hard to reflect fully the characteristics of FVV in the system [16]. However, it didn't consider about the zoom-in/out function which is one of the supporting parts of FVV service. One of the studies suggested the FVV streaming system in detail, but it focused on the depth video for the synthesized view [17]. One of other studies suggested user view-oriented video services which also considered zoom in/out function, but this paper has not specific example for the method and deep consideration about FVV [18]. There was a VR transmission study, which has similar point of FVV, but it considered on sending over MMT [19].

Thus, in this paper, we considered zoom in/out view streaming based on object information of FVV to meet the requirements and fulfill the satisfaction of users. This paper suggests streaming method that considered user side to provide FVV video. This method can keep the quality and resolution of the video watching. It can cause some video request delay because of calculation for video selecting and it seems to be complicated compared to existing method. However, with this method user didn't have to move zoom in area manually and can watch the best zoom-in view that supports the object view that user wants to watch optimized.

3. A Proposal for Zoom-in/out View Streaming based on Object Information of Free Viewpoint Video

In this chapter, we design the zoom in/out function so that the object can be effectively viewed in detail in FVV service streaming. In FVV service, the viewpoint may be moved or fixed according to the movement of an object in the video. It defines the range of zoom in/out and analyzes the video information to compose it. By designing a method so that the configured information can be provided together with the video stream, a zoom in/out function suitable for FVV can be made.

3.1 Definition of Zoom In/Out Range of FVV

FVV service includes a function for zoom in/out in the service. By defining within the service, not the individual player's function, it is specified that the zoom in/out function must be considered essential within FVV service. To this end, regardless of the performance of each player or the user service environment, it is necessary to design so that the same zoom in/out distance and service quality can be provided when the same content is consumed. We set the range of the viewpoint in the points that it is difficult to calculate the extent of the range according to the movement of the object and that one content needs the same zoom in/out distance. Also, in order to specify the cropping position and size of the zoomed in/out screen, this paper proposes to use object information.

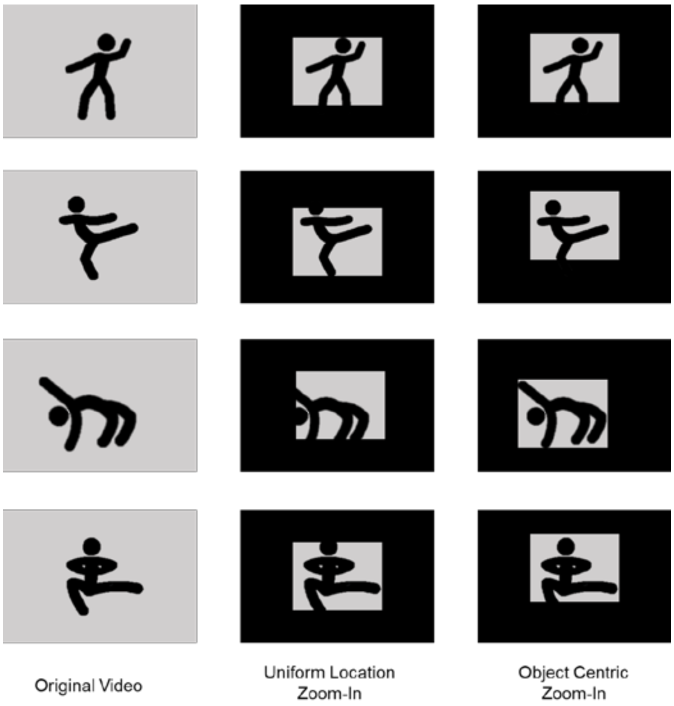


Fig. 5. Comparison of uniform location zoom-in and object centric zoom-in methods

Fig. 5 shows the comparison between the uniform location zoom-in (general methods) and object centric methods which is the proposed methods. There may be cases where zoom in of a general size and location does not sufficiently show the object to be viewed. However, when focusing on the object, the important scene of the object should be guaranteed as much as possible. However, object location information such as coordinate values of the object must be available. In this paper, we propose methods using object coordinate values.

If the object is moving, the appearance of the object shown in the video at the current time may not be the best. In that case, you should change it to a video that can show the object in as much detail as possible and then play it. The advantage of FVV is that it can transmit and process multiple videos to show video from various viewpoints, but since the video cannot be sent indefinitely, meaningful videos must be selected. Therefore, in this paper, we propose a method of selecting, receiving, and zooming in/out a video based on an object.

3.2 Design of Zoom In/Out Streaming Method of FVV

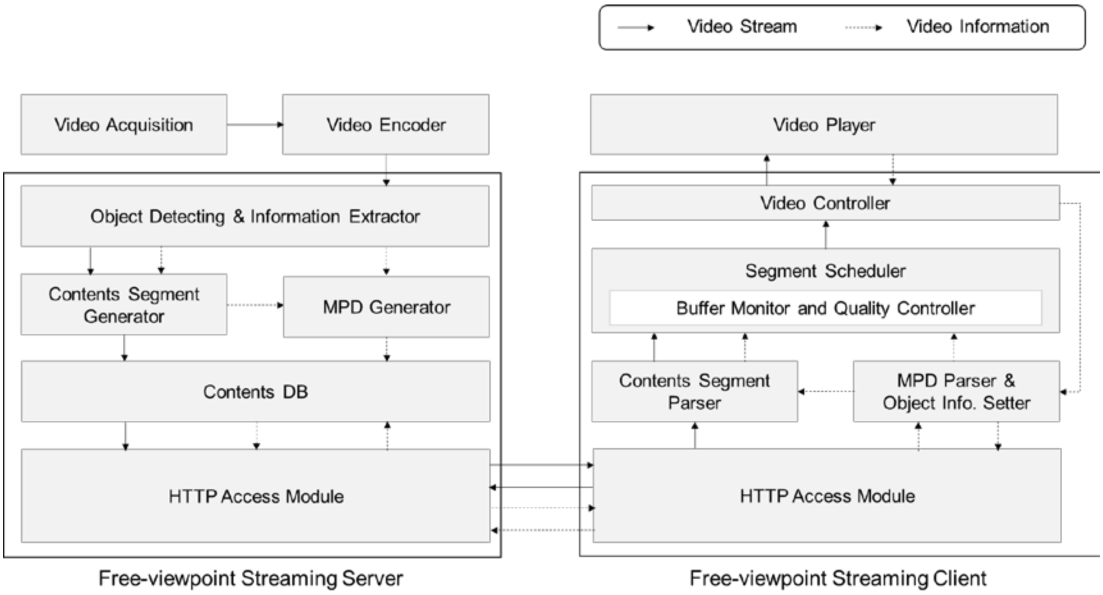

Video Stream
Video Information
Video Acquisition
Video Encoder
Object Detecting & Information Extractor
Contents Segment Generator
MPD Generator
Contents DB
HTTP Access Module
Free-viewpoint Streaming Server
Video Player
Video Controller
Segment Scheduler
Buffer Monitor and Quality Controller
Contents Segment Parser
MPD Parser & Object Info. Setter
HTTP Access Module
Free-viewpoint Streaming Client

Fig. 6. Suggested Free-viewpoint video streaming system

In order to provide the zoom-in/out streaming method proposed in this paper, it is necessary to establish a system as shown in **Fig. 6**. Because it is necessary to prepare to stream segmented video based on object information and to deliver object-related information to the user who requested the service in advance.

When an encoded video is input after video acquisition, the content server recognizes an object and extracts object information. Object information is generated for each frame, and the generated object information must be included in accordance with a segment, which is a transmission unit of video. In the contents segment generator, object-related playback time information is included, and it is very important to determine the segment size. Since all videos cannot be provided due to the characteristics of FVV, the optimal amount should be determined for each video. In this paper, to determine the segment size of an object, the optimal transmission amount for each video is calculated according to the size of the video after the entire video is captured. In order to match the optimal amount of video transmission, the section in which the video including object information is selected is first checked, and the segment length is determined while adjusting the number of selectable videos for each segment according to a specific criterion. The algorithm and selection method are being studied separately, and it is not fully treated in this paper because it is too complicated topic to be handled lightly. So, methods to provide and optimally deliver zoom-in information within the already selected segment length is focused and studied. In order to compose a video in the MPD generator, information on the type and number of objects is provided from the object information extractor. In addition, information such as which object information each segment includes and whether object-related reproduction is possible through a combination of segments can be additionally provided from the segment generator. All information is uploaded to the content DB and provided to users and clients through HTTP.

The client first requests the MPD. After MPD is received, object information is first acquired and ready for video reception. Input the received video information to the content segment parser. The segment is divided into video data and object-related data in the process of being parsed, and information is managed. The information is transmitted to the segment scheduler together with the object information in MPD document, and the received video is scheduled to be played based on the object information. The segment is provided to the video controller for the user to view. Through this system, we designed an object to be selected and to manage object information for video zooming.

On the other hand, it is very important in zoom in to determine the object coordinate information of the video. In the case of FVV, the displayed video on user side is changed and moved within the content so that it can be changed to a viewpoint video including the optimal object appearance according to the movement of the object.

Therefore, the moving view first performs object recognition for each view video. Thereafter, a view video including an object section in which a specific object is most visible is selected as one stream.

This selection method may request various video selections for small section reproduction. However, in this paper, we aim to provide stable zoom in/out while providing an optimal video stream that reproduces the object at the center. Segments with the same time information are requested for the number of videos according to the transmission capacity of each user, and object coordinates of the same size are designated in the same segment reproduction section.

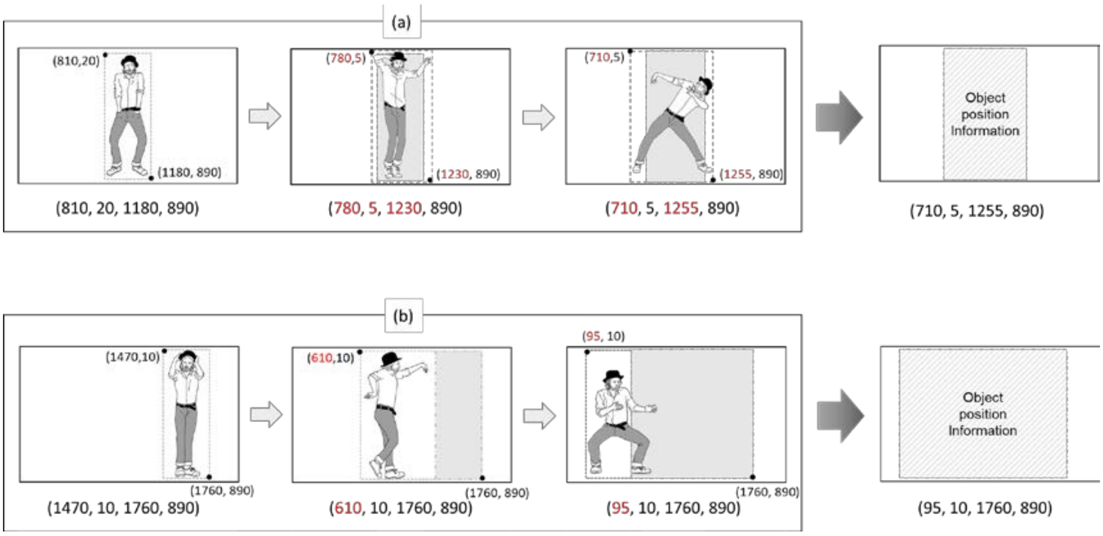


Fig. 7. Example of getting x, y coordinate values of objects in segment

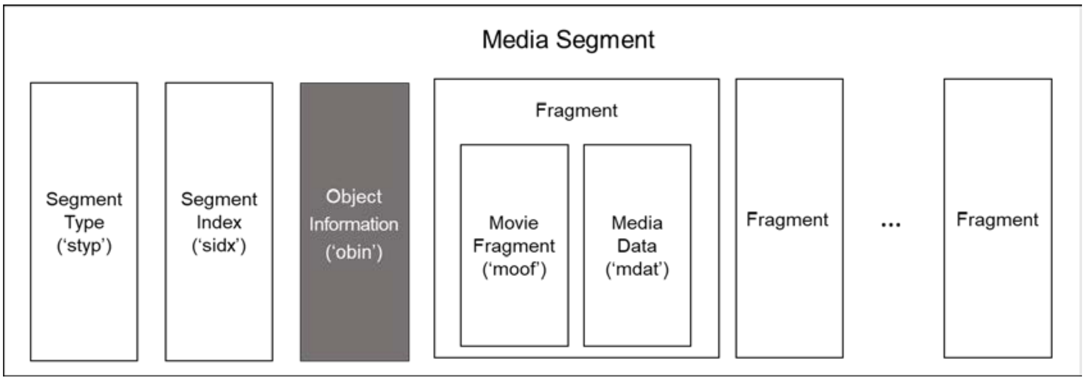


Fig. 8. ISOBMFF media segment structure including suggested object information

Once the segment size has been determined, it is necessary to determine only one optimal object coordinate information for each segment. The coordinate information of the object is different for each frame, and providing all information is because not only is providing too much information, but also changing the zoom in size of the object for each frame reduces the immersion.

Fig. 7 shows an example of determining object coordinate values. In case of (a), it is example of little movement of the object. In this case, the size coordinates of the object are determined by updating the x and y coordinate values. The case of (b) means that there is a lot of movement of the object. Depending on the movement of the object, the coordinates of the object displayed on the screen may vary. Since more space is set in (b) than in (a), the object setting size may be smaller than the actual size of the reproduced object. In this paper, we aim to guarantee the shape of the object as much as possible without the part of the object being hidden or disappearing.

This paper proposes to use ISOBMFF media segment structure to provide object information values. Since it is decided in this paper to use one coordinate value per segment, it is efficient to use the segment box format that contains the necessary information while one segment is consumed. However, since the information defining the coordinate information of the object or the reproduction time information does not exist in the existing box type, it is proposed to define an additional box type and configure the necessary information.

Fig. 8 shows the media segment composition including the proposed object information box. ObjectInformationBox('obin') is defined to contain object information. Before video fragment decoding is performed, information related to an object may be obtained through a box in advance, and decoding may be prepared based on the object information. **Fig. 9** shows the box structure in which object information ('obin') for FVV service proposed in this paper is written in ISOBMFF format. Through the 'obin' box, the number of objects in one segment and the existence of objects are checked through the unique identification value for each object. Then, the coordinate information required for zooming in/out of the object is used.

Each segment corresponds to one video stream, and not only one segment is acquired for decoding optimized for object viewing, but all the same duration segments of multiple views included in the same *Representation* are acquired. The main field names and definitions of 'obin' box are as follows.

```
aligned(8) class ObjectInformationBox extends Box('obin', contents_quality, 0) {
        unsigned int(32) object_count;
        for(j=0; j<object_count; j++) {
                unsigned int(32) object_id;
                if(video_dependency==0) {
                /* case of Independent video */
                         unsigned int(32) x_axis;
                         unsigned int(32) y_axis;
                         unsigned int(32) x_width;
                         unsigned int(32) y_height;
                }
                if(contents_quality==0) {
                /* case for low network capacity */
                         unsigned int(32) object_count_l;
                         for(k=0;k<object_count _l;k++) {
                                  unsigned int(32) start_frame;
                                  unsigned int(32) last_frame;
                         }
                }
                else if(contents_quality==1) {
                /* case for high network capacity */
                         unsigned int(32) object_count_h;
                         for(p=0;p<object_count _h;p++) {
                                  unsigned int(32) start_frame;
                                  unsigned int(32) last_frame;
                         }
                }
        }
}
```

Fig. 9. Suggested 'obin' box structure for object information

- object_count: It means the number of objects that contain information in the relevant segment, and may not match the total number of objects in the content.
- object_id: It means the unique identification ID of an individual object within one content, and the same value is maintained in all segments.
- video_dependency: It means whether the corresponding segment video is an independent video. If the value is set to 1, it means dependent cases such as split screen of video.
- x_axis: It means the starting (left x value) coordinate of the object in the corresponding segment. If the size required by the display to be played is different from the basic size of the content expressed in MPD, the value of x_axis is also changed by the corresponding ratio.
- y_axis: It means the starting (upper y value) coordinate of the object in the corresponding segment. If the size required by the displayed display is different from the basic size of the content expressed the MPD, the value of y_axis is also changed by the corresponding ratio.
- x_width: It means the width (right x value) of the object in the segment. If the size required by the display to be played is different from the basic size of the content

expressed in MPD, the value of x_width is also changed by the corresponding ratio.

- y_height: It means the length of the object in the corresponding segment (lower y value). When the size required by the display to be reproduced is different from the basic size of the content expressed in MPD, the value of y_height is also changed by the corresponding ratio.
- object_count_l: When the bandwidth is narrow, it means the number of times the corresponding object is displayed on the screen within one segment. It may be different from the case where the bandwidth situation is good or wide.
- object_count_h: When the bandwidth is wide, it means the number of times the corresponding object is displayed on the screen within one segment. It may be different from the case where the bandwidth situation is bad or narrow.
- start_frame: It means the frame number at which the section where the object is expressed starts.
- last_frame: It means the frame number at which the section where the object is expressed ends.

Although this paper does not deal much with streaming techniques, it is necessary to consider MPD with streaming path in order to transmit object-oriented video. The values for the segment path are listed in MPD so that the object information is first obtained and the receiver can request the video. However, it is difficult to compose information about an object with the existing MPD writing method. In this paper, we propose a method that maintains MPD configuration format, but configures the service structure based on object information and transmits it adaptively to the network environment when the client selects object information to provide the service.

Fig. 10 shows an example of the proposed object-oriented MPD structure. If the existing *AdaptationSet* means a set encoded in several alternative ways, such as video, audio, and subtitles, it is configured to request multiple video sets for one object instead of one video. And the list of segments can be managed with *SegmentList* in *Representation*. Through *Representation*, the number of configured videos can be managed differently. However, in this paper, network conditions such as the user's transmission capacity are not considered, so it is configured in a single format.

The system and transmission method for zoom in proposed in this paper seems to be complicated compared to existing method. However, the reusability is very high in that the free-viewpoint video is optimized for the screens of various users in advance and can be provided in consideration of the transmission capacity. In addition, by allowing the client to decide, the burden on the server can be reduced, and the viewing immersion can be enhanced by the convenience of viewing the user without any hassle and without directly adjusting the desired object.

```
<?xml version="1.0"?>
<MPD xmlns=“urn:mpeg:dash"schema:mpd:2011" profiles=“urn:mpeg:dash:profile:full:2011">
<Period>
  <baseURL>MultiviewContents/</baseURL>
  <AdaptationSet mimeType="video/m4s">
    <baseURL>object1/</baseURL>
    <Representation id="obj1" bandwidth="25600000" width="3840" height="2160">
        <SegmentList timescale="90000" duration="5400000">
          <RepresentationIndex sourceURL="object1.sidx"/>
          <SegmentURL media="object1_1.m4s"/>
          <SegmentURL media="object1_3.m4s"/>
          <SegmentURL media="object1_4.m4s"/>
          <SegmentURL media="object1_5.m4s"/>
          <SegmentURL media="object1_7.m4s"/>
        </SegmentList>
     </Representation>
  </AdaptationSet>
  <AdaptationSet mimeType="video/m4s">
   <baseURL>object2/</baseURL>
    <Representation id="obj2" bandwidth="25600000" width="3840" height="2160">
        <SegmentList timescale="90000" duration="5400000">
                    …   omit   …
        </SegmentList>
     </Representation>
  </AdaptationSet>
```

Fig. 10. Example of proposed object-oriented MPD document

4. Experimental Results

In order to verify the proposed method, we compared it with the existing method of zooming in/out of video. There are not many videos in FVV that are currently publicly opened and available, so we prepared a video that was filmed separately. The selected FVV is an image of a total of 6 dancers, and there is a change in viewpoint according to the movement of an object in the video.

First, video was acquired to fit the experiment. In the case of FVV, a total of 15 cameras were installed and acquired, and the distance between the cameras was 40 cm, and efforts were made to minimize errors from the shooting environment of each camera by maintaining the same distance. For the experiment, only the video section of 60 seconds was selected and used. After acquiring the image, it is necessary to obtain video section information by applying object recognition to the video. The technology applied to the object recognition method considers the method of applying AlphaPose. A tool called AlphaPose was used to continuously maintain the recognition of the movement of the object. Object and face recognition can be performed through AlphaPose, but the recognition technology does not even provide object identification information. For this reason, the Face Recognition tool was additionally referred to. Since it is a tool to which the standard for judging the identity between objects is applied, the corresponding value was used to obtain recognition information for each object and used in the experiment. Because the size of a segment is important for streaming, it is necessary to adjust the spacing. However, since the method of controlling the amount of video transmission does not control only object information and requires consideration of other factors, it is not covered in this paper. **Table 2** shows the information of FVV encoding option. **Fig. 11** is a series of FVV streams selected through object information. One object was arbitrarily selected, and the selected FVV stream was extracted based on the corresponding object information. **Fig. 12** means a video section corresponding to a total of 60 seconds in a total of 15 videos. All the viewpoint videos in which sections with a high object recognition rate exist among 30 frames corresponding to 1 second are displayed. It can be seen that there are sections in which the position of the object is fixed, but there are also cases in which the recognition rate is high at various viewpoints within 1 second due to a large motion change. In this case, it is possible to limit the number on the server side, or use the Representation tag in the proposed MPD document to configure and manage the minimum version for the case where the transmittable amount is small and the maximum version for the case where the transferable amount is large. There are sections that require a lot of videos like the fourth section, but you can see that most of them require 2-3 videos. In addition, due to the nature of the video used in the experiment, a video with a slight distance from the video views that are closely attached is sometimes selected, so it can be inferred that the required video interval is affected by the shooting conditions, and for each video, the It can be seen that in order to watch the video, a differentiated method is required rather than a lump sum.

As shown in **Fig. 13**, the zoom in/out result centered on the object can also confirm the meaningful result. For comparison with the existing method, 2x digital zoom was set in the existing method, and only the corresponding area was zoomed in in the crop method based on the coordinate information obtained after object recognition. For #1, since the height of the object is similar to the height of the screen, the effect of the proposed zoom in is not great. However, in the case of the existing method, since the center is zoomed in by 2x, the location where it is difficult to check important information such as the face of the object is shown. In case of #2, since the object comes out small, zoom in is meaningful when viewing the object to be viewed located at the center based on the coordinate information of the object. In the existing method, although an object to be viewed appears, it is located at the edge. In #3, the object can be seen clearly in both the existing and the proposed method, but there is a section where a part cannot be seen in the existing method, whereas the proposed method shows the object completely.

Table 3. Encoding option

Item	**Value**
resolution	3840x2160
codec	h.265 (HEVC)
crf	22
fps	30

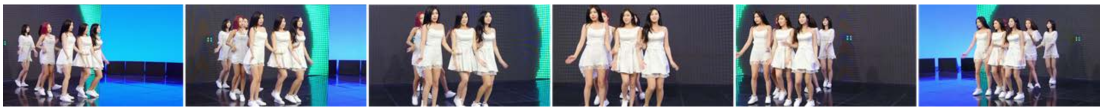


Fig. 11. Example video for experimentation

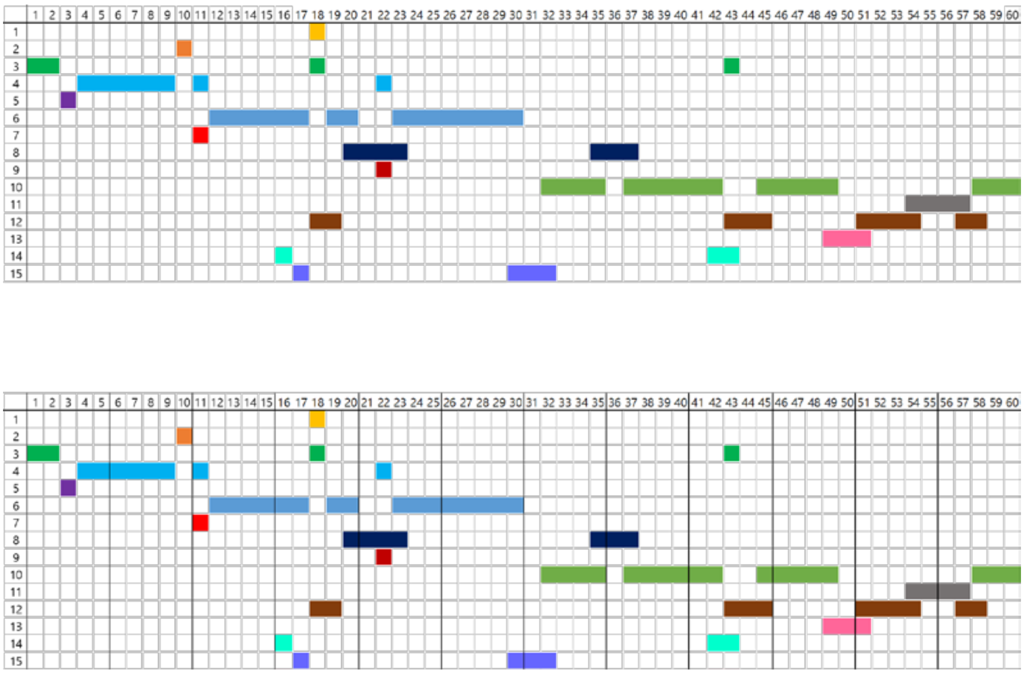


Fig. 12. Stream of Free-viewpoint video selected based on object information

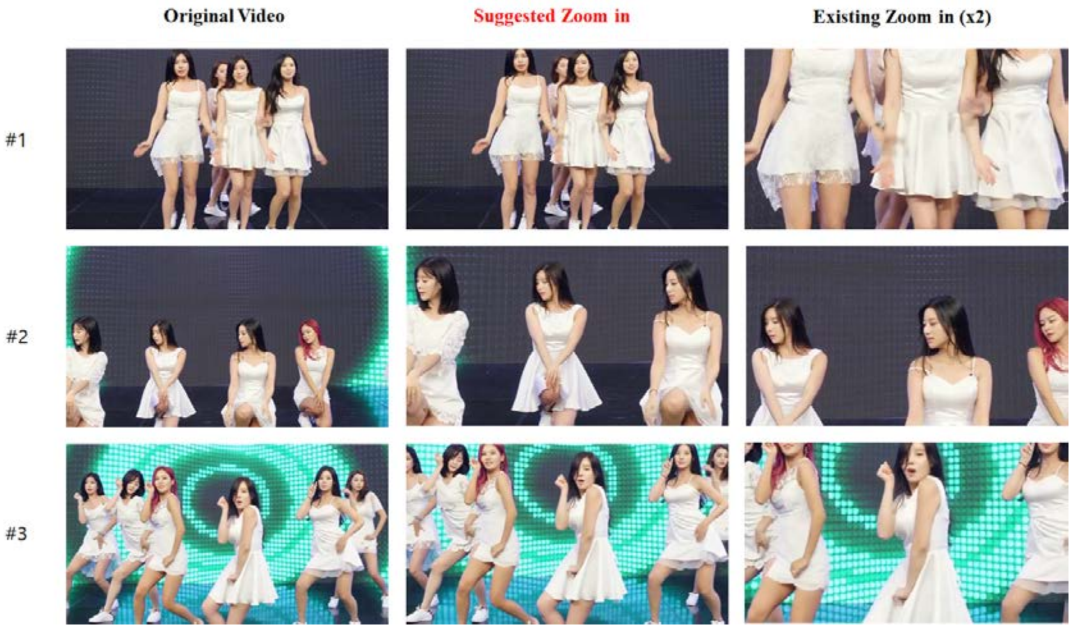


Fig. 13. Video result of experiments

5. Conclusion

Recently, consumers' demand for active video viewing is increasing, and in particular, there are many cases where they only want to watch a specific part of the stage. The various viewpoint directions and zoom in/out functions provided by FVV service can reflect these needs, so the importance is expected to grow gradually. However, since FVV includes a plurality of videos, changes may occur frequently due to movement of the viewpoint. Therefore, it is necessary to provide a plurality of videos while maximally guaranteeing a user's desired viewing.

In particular, providing an object-oriented FVV service is also important for zoom in. When video zooming in in the usual way, it cannot be guaranteed to zoom in around the object. Although the zoom in/out function is included in the definition of FVV, the details of how to apply it are not considered. However, in order to satisfy the viewing satisfaction of users who want more realistic images and interactive viewing than before through FVV viewing, it is important to bind the zoom-in method and the viewing target.

To solve this problem, this paper proposes a zoom in/out technique of object-centered dynamic adaptive streaming of FVV. In the proposed technique, we proposed a method of constructing playback section information by searching for optimal recognition scenes for each object among video from multiple viewpoints, and extracting object information. In order to maximize the video service to be viewed based on the object, the video section information in which the object is optimally recognized was considered. We proposed a method of selecting and requesting a video based on an object, and providing coordinate information so that zoom in/out can be made based on the provided video. In this paper, we aimed to guarantee the shape of the object as much as possible without the part of the object being hidden or disappearing.

ISOBMFF media segment structure is used to provide object information values. Since information defining object coordinate information or playback time information does not exist in the existing box type, an ObjectInformationBox (‘obin’) that defines an additional box type and composes necessary information is proposed. Before video fragment decoding is performed, information related to an object may be obtained through a box in advance, and decoding may be prepared based on the object information.

Although this paper does not deal much with streaming techniques, it is necessary to consider MPD with streaming path in order to transmit object-oriented video. While maintaining MPD configuration format, MPD structure is additionally proposed to provide a service by configuring the service structure based on the object information and adaptively transmitting it to the network environment when the client selects the object information. We tried to confirm the significance of the application of the proposed techniques through simulation. In the existing method, there is a section where a part cannot be seen, while the proposed method confirms that the object is fully visible. Through the proposed content, the user can zoom in FVV centering on the object. In future research, we intend to study the method of screening video considering the network situation, especially optimal transmission amount.

References

[1] “How Canon's Free Viewpoint Video System brings a new perspective to Rugby World Cup 2019™ action”. [Online]. Available: https://www.canon.co.uk/pro/stories/free-viewpoint-video-rugby-world-cup/

[2] M. Tanimoto, M. Panahpour Tehrani, T. Fujii and T. Yendo, "FTV for 3-D Spatial Communication," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 905-917, April 2012. Article (CrossRef Link).

[3] M. Tanimoto, "FTV (Free-viewpoint TV)," in *Proc. of 2010 IEEE International Conference on Image Processing*, pp. 2393-2396, 2010, Article (CrossRef Link).

[4] Gooman Park, “Free viewpoint content composition technology development case,” *broadcasting and media*, Vol. 24, No. 3, pp. 24-34, Aug. 2019. Article (CrossRef Link).

[5] Gooman Park, "Free viewpoint video technology trends and development examples of extended free-view point of view," *Information and Communication*, Vol. 36, No. 12, pp. 3-9, Nov. 2019. Article (CrossRef Link) .

[6] "Free viewpoint video system using canon's unrivaled imaging technologies". [Online]. Available: https://global.canon/en/technology/frontier18.html.

[7] G.S. Lee, J.Y. Jeong et al., "Standardization Trend of 3DoF+ Video for Immersive Media," *Electronics and Telecommunications Trends*, Vol. 34, No. 6, pp.156-163, Dec. 2019. Article (CrossRef Link).

[8] Sid Yingze Bao, Yu Xiang and Silvio Savarese, "Object Co-detection," in *Proc. of European Conference on Computer Vision*, pp.86-101, Oct. 2012. Article (CrossRef Link).

[9] J. Gwak, G. Park and M. Jeon, "Viewpoint Invariant Person Re-Identification for Global Multi-Object Tracking with Non-Overlapping Cameras," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 4, pp. 2075-2092, 2017. Article (CrossRef Link).

[10] "face_recognition package". [Online].
Available: https://face-recognition.readthedocs.io/en/latest/face_recognition.html

[11] N. Kim and B. Lee, "Analysis and Improvement of MPEG-DASH-based Internet Live Broadcasting Services in Real-world Environments," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 5, pp. 2544-2557, 2019. Article (CrossRef Link).

[12] Robert Seeliger, Daniel Silhavy, Stefan Pham and Stefan Arbanowski, "Cross-platform ad-insertion using HbbTV and MPEG-DASH," in *Proc. of Asia Pacific Conference on Multimedia and Broadcasting*, pp.7-13, Nov. 2016. Article (CrossRef Link).

[13] EasyTV: Easing the access of Europeans with disabilities to converging media and content, "Content adaptation using DASH streaming services," Oct. 2018. [Online]. Available: https://easytvproject.eu/files/D4.4.pdf

[14] ISO/IEC 23009-1:2014 Information Technology—Dynamic adaptive streaming over HTTP (DASH)—Part 1: Media presentation description and segment formats. [Online]. Available: https://www.iso.org/standard/65274.html

[15] Ingo Kofler, Robert Kuschnig and Hermann Hellwagner, "Implications of the ISO base media file format on adaptive HTTP streaming of H.264/SVC," in *Proc. of 2012 IEEE Consumer Communications and Networking Conference (CCNC)*, pp.549-553, Jan. 2012. Article (CrossRef Link).

[16] Ahmed Hamza and Mohamed Hefeeda, "A DASH-based Free Viewpoint Video Streaming System," in *Proc. of Network and Operating System Support on Digital Audio and Video Workshop*, pp. 55-60, Mar. 2014. Article (CrossRef Link).

[17] Minjae Seo and Jong-Ho Paik, "Implementation Method for DASH-based Free-viewpoint Video Streaming System," *Journal of Internet Computing and Services*, Vol. 20, No. 1, pp. 47-55, 2019. Article (CrossRef Link).

[18] Minjae Seo and Jong-Ho Paik, "Bandwidth-Efficient Transmission Method for User View-Oriented Video Services," *Computers, Materials & Continua*, Vol. 65, No. 3, pp. 2571-2589, Sep. 2020. Article (CrossRef Link).

[19] J. Lee, J. Lee, J. Lim and M. Kim, "Bandwidth-Efficient Live Virtual Reality Streaming Scheme for Reducing View Adaptation Delay," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 1, pp. 291-304, 2019. Article (CrossRef Link).

Minjae Seo received the B.S, M.S. and Ph.D. degrees in Computer Science from Seoul Women's University, Korea, in 2013, 2016, and 2021 respectively. From 2021, she works at STANS Inc. as a general manager. Currently, her major research interests are FVV service and streaming service like DASH, MMT. she is also interested in Digital-twin system and Metaverse services.

Jongho Paik received the B.S., M.S., and Ph.D. degrees in the school of Electrical and Electronic Engineering from Chung-Ang University, Seoul, Korea, in 1994, 1997, and 2007 respectively. He was a Director with Advanced Mobile Research Center at Korea Electronics Technology Institute (KETI) by 2011. Since 2011, he is currently an assistant professor in the department of multimedia, Seoul Women’s University, Seoul. His research interests are in the areas of web-based communication, software testing, video communications system design and system architecture for realizing advanced digital communications system and for advanced mobile broadcasting networks as well.

Gooman Park received a Ph.D. and a M.S. in Electronics Engineering from Yonsei University, and a B.S. in Electronics Engineering from Hankook Aviation University. He is a professor of Department of Electronics and IT Media Engineering at Seoul National University of Science and Technology, Seoul, Korea. He was a senior engineer at Samsung Electronics, Suwon, Korea. His current research interest includes computer vision, immersive media.