TOWARD TELE-EXPERIENCE:

ENHANCED VIEWING EXPERIENCE BY SYNCHRONIZED UHDTV AND FREE-VIEWPOINT AR

Y. Kawamura, T. Kusunoki, Y. Yamakami, H. Nagata and K. Imamura

NHK (Japan Broadcasting Corporation), Japan

ABSTRACT

The integration of ultra-high-definition television (UHDTV) broadcasting and augmented reality (AR) content through broadband Internet presents an enhanced viewing experience that has evolved on a different axis than quantitative specifications. Therefore, we propose a new viewing style where viewers watch a UHDTV program on television while simultaneously experiencing free-viewpoint AR on their own mobile devices.

In this paper, we describe an object-based real-time streaming system of dynamic three-dimensional objects for AR played on mobile devices such as smartphones and tablets. Each component object synchronized with a UHDTV program is transmitted along with a unique object identifier in transport packets over broadband Internet. Consequently, terminal devices or network nodes use the object identifier to filter required objects based on individual viewports. Furthermore, an object is transmitted using multiple representations in different resolutions and bitrates to be selected according to the viewing distance from the object. We conducted a transmission experiment to evaluate the packet level filtering of required objects in appropriate resolution.

INTRODUCTION

4K/8K ultra-high-definition television (UHDTV) satellite broadcasting in Japan was commercialized in December 2018. It is the world’s first practical 8K broadcasting, which introduces viewers to the world of immersive video and audio. 8K broadcasting is regarded as the ultimate two-dimensional (2D) video medium. In other words, consumers will probably prefer qualitative diversification of viewing on television to having a television with quantitatively higher specifications. From a technical perspective, one of the next challenges for future digital media technologies will be to deliver three-dimensional (3D) information.

Recently, augmented reality (AR) and virtual reality (VR) technology has attracted increased attention. 360-degrees VR video is already common among over-the-top video streaming services because VR videos are streamed using the same technology used for 2D video streaming. On the other hand, the application of AR is still limited. For examples, AR is used in computer games featuring imaginary creatures and virtual fitting of furniture. There are,

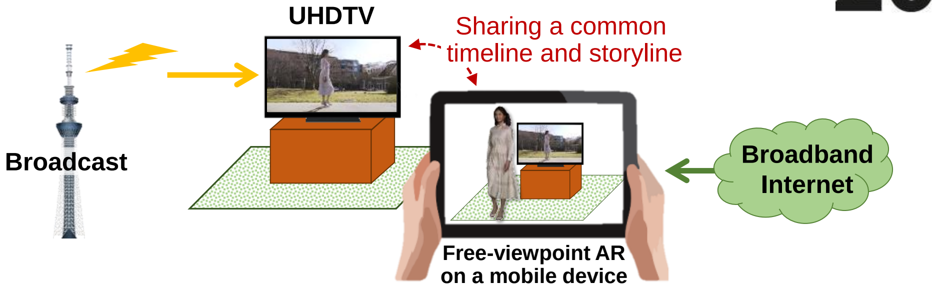

UHDTV
Sharing a common timeline and storyline
Broadcast
Broadband Internet
Free-viewpoint AR on a mobile device

Figure 1 – Concept of the integration of UHDTV and AR

however, few AR contents with timelines and stories like movies or TV programs. Until now, AR experience is limited to just placing static objects in an augmented environment. We believe that AR has the potential to enhance the experience of live content consumption and human communication. However, there has been insufficient research on distribution technology for AR contents in order to utilize and maximize the potential.

Based on these backgrounds, we have proposed a new viewing style where viewers watch a UHDTV video on television while simultaneously experiencing 6 degrees of freedom (6DoF) AR on their own mobile devices. Figure 1 illustrates our concept of the integration of UHDTV and AR, which provides volumetric tele-existence of performers and spatial experience bringing viewers into the content world. The UHDTV and AR share a common timeline and storyline; therefore, the AR experience is not just placing static objects.

In the following sections, we describe our distribution model and prototype content that embodies this concept, the detail of our implementation including object-based transmission mechanisms, and a fundamental experiment of the object-based transmission.

ENHANCED VIEWING EXPERIENCE OF UHDTV WITH SYNCHRONIZED AR

Figure 2 shows our distribution model of synchronized UHDTV program and AR content, which is based on an extension of UHDTV broadcasting. As shown in the upper part of Figure 2, video and audio are delivered over a normal UHDTV broadcast. Then, 3D objects linked to the broadcast program are delivered in real-time through broadband Internet, as shown in the lower part of Figure 2. 3D objects are displayed by AR on a mobile device through its camera. The AR is rendered and displayed at synchronized presentation timing with the video and audio on the television.

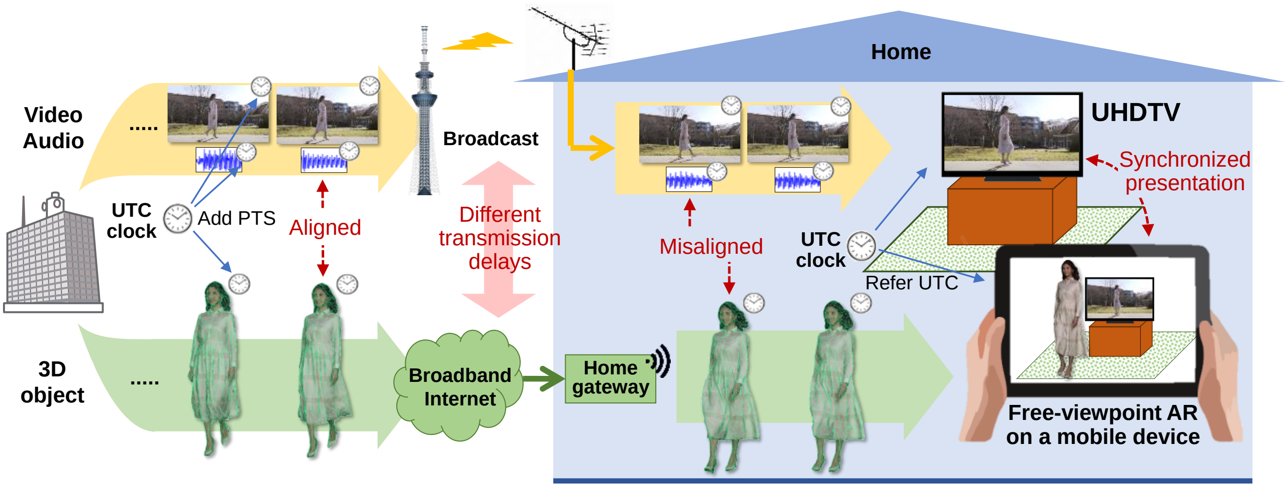


Figure 2 – Distribution model of synchronized UHDTV program and AR content

The synchronized UHDTV program and AR content enhances viewing experiences. The UHDTV program provides ultra-high definition visual images but offers only one view for all viewers. The view represents the intension of its program director. On the other hand, AR provides volumetric tele-existence of 3D objects. Although the visual quality is limited based on available technologies, it offers infinite views for all individual viewers, according to personal preferences. Viewers can manipulate the viewing experience as though they were operating cameras by themselves. For example, a television star who ordinarily can only be seen on a television screen is instantly drawn into their familiar environment. Thus, the viewers share the same space with the star beyond the frame of the television.

In order to provide such a service for sports programs or other live contents it is required to realize real-time transmission of 3D object data of changing shapes as the program progresses. Video and audio over broadcasting and 3D objects over broadband Internet have different transmission delays. Therefore, to match the timing during playback, timing information is applied to the content as it is transmitted in real-time. A data frame of a 3D object is transmitted along with a Coordinated Universal Time (UTC) based presentation timestamp (PTS) in the same way as MPEG media transport (MMT). MMT standardized by ‘ISO/IEC (1)’ is an Internet Protocol-based media transport scheme adopted for the UHDTV satellite broadcasting in Japan. At the mobile device, the PTS is used to display the frame to synchronize with the video frame displayed on the television, as described in ‘Kawamura et al (2).’

Prototype Content Production

We created a prototype content that embodies the proposed viewing experience using available technologies. We used an existing volumetric capture studio to produce the synchronized UHDTV and live-action AR content. Several systems currently exist, including 4DViews, Intel Studios, and Microsoft Mixed Reality Capture Studios described in ‘Collet et al (3)’. These systems commonly match image features from multiple cameras surrounding a performer and generate sequential volumetric data at a fixed rate, such as 30 frames per second (fps), which is similar to frames of a 2D video. These studios are painted green or covered with green curtains to make cutting out the shooting object easy. The green background will become unnecessary with future technological advances.

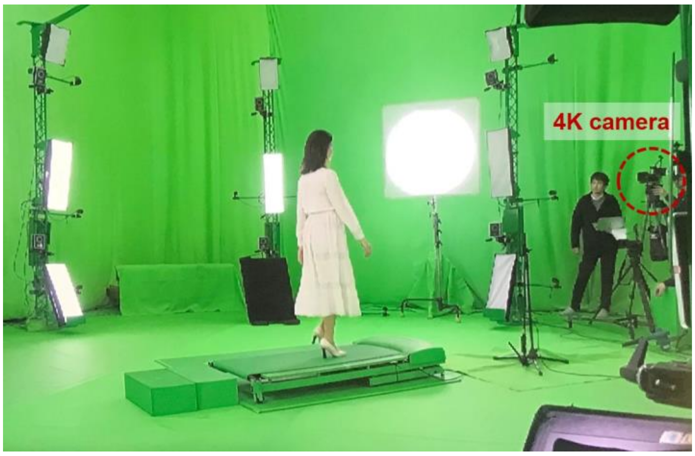


Figure 3 – Simultaneous shooting of 3D volumetric and 4K

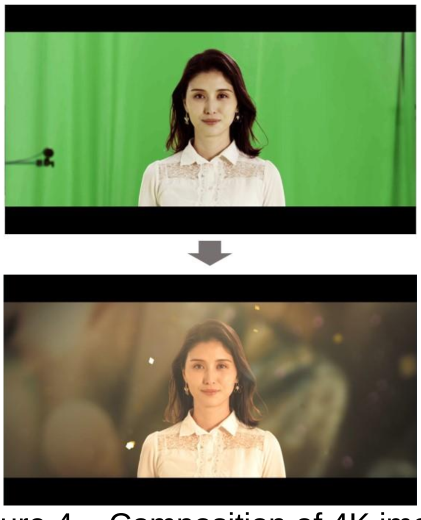


Figure 4 – Composition of 4K image

Also, we placed a 4K camera in the volumetric capture studio to shoot the object in 4K video simultaneously with the volumetric sequence, as shown in Figure 3. Since the background of the 4K image is also green, we synthesize a background image for the 4K video, as shown in Figure 4. Figure 5 shows the demonstration of the prototype content, in which Ms. Manami Hashimoto, a famous Japanese actress, appeared as a performer. The actress seamlessly connected the space inside and outside of the TV by interlocking high-definition images on a large UHDTV screen and AR with all-around free-viewpoints. Figure 6 shows the scene wherein the performer, who was inside the video on the UHDTV, came out of the UHDTV. The high-precision synchronization of UHDTV video and AR content enables the seamless viewing experience between UHDTV and AR.

TRANSMISSION AND RECEPTION OF 3D OBJECTS

Figure 7 illustrates a block diagram of the data processing path from acquisition to presentation of 3D objects. We explain the data framing, packet encapsulation, AR player application on a mobile device, and object filter at a network node.

Data Framing

The original format of volumetric data depends on the capture system. However, it should be delivered to consumer devices in an open standard format, which is not restricted to an application or vendor. We used OBJ described in ‘Wavefront (4)’ as the interface format to

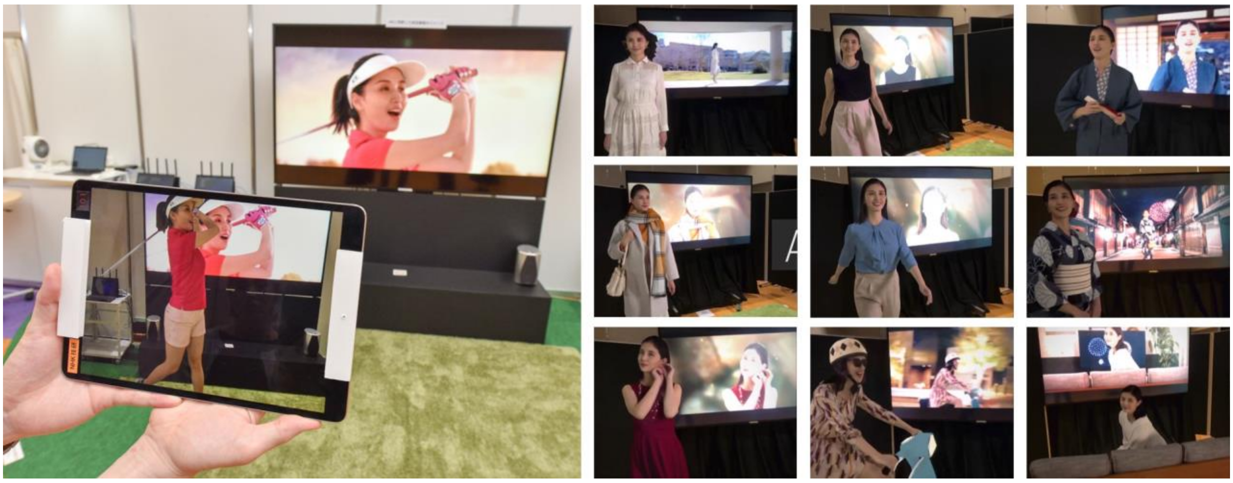


Figure 5 – Prototype content featuring Ms. Manami Hashimoto

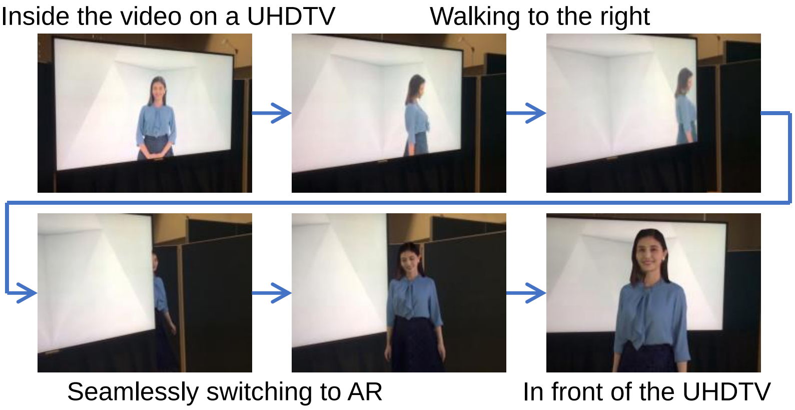


Figure 6 – Seamless coming out from UHDTV to AR

a content server because OBJ is a widely used format for 3D models, which consist of geometry data and texture image. As for texture images, we used JPEG format.

Before streaming, geometry data and texture image for each frame are merged into a combined binary data frame. Since OBJ (.obj) files describe geometry data using ASCII text, it is not efficient to transmit the OBJ file as it is. Instead, we designed a binary data structure that is equivalent to the original OBJ file. Figure 8 shows a descriptive example of volumetric data, and Figure 9 shows the structure of the combined volumetric data frame. 3D coordinates (X, Y, Z) and texture-mapping coordinates (U, V) of each vertex are quantized as 16-bit floating-point values (Float16). The vertex indices for each face are represented as unsigned 16-bit integers (Uint16). Although an original volumetric data may have vertex vectors, they are omitted to reduce the data to be transmitted.

There is a trade-off between visual quality and transmission bitrate for 3D objects. The number of vertices and faces in geometry data are among the parameters that drive the trade-off. The other parameters are the resolution and compression ratio of the JPEG-compression for the texture image. Figure 10 shows examples of visual quality rendered on the same presentation scale using geometries at different numbers of vertices and textures at different resolutions and compression ratios listed in Figure 11.

Packet Encapsulation

Each data frame is divided into fragments, multiplexed into User Datagram Protocol over Internet Protocol (UDP/IP) packets, and transmitted over the Internet in real-time. At the top of UDP payload is the header, which includes packet identifiers (PIDs). The PIDs are used to multiplex signalling information and multiple sequences of volumetric data into a single UDP/IP flow. Packets whose PID equals 0 delivers signalling information as the entry point

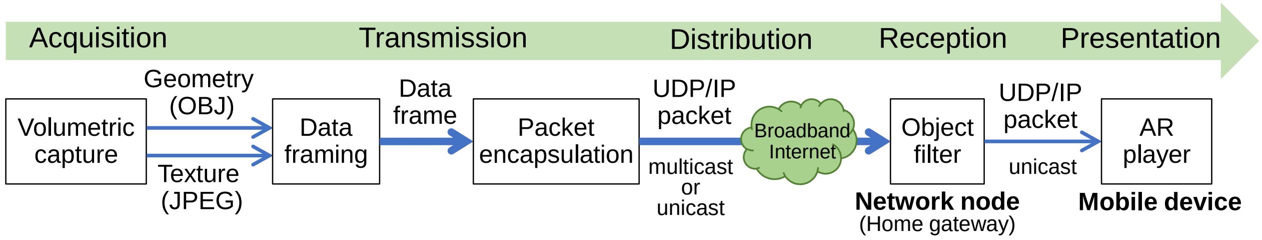


Figure 7 – Block diagram of data processing path

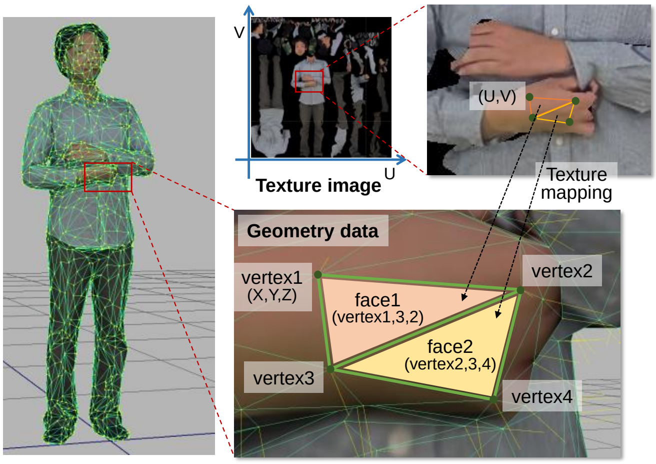


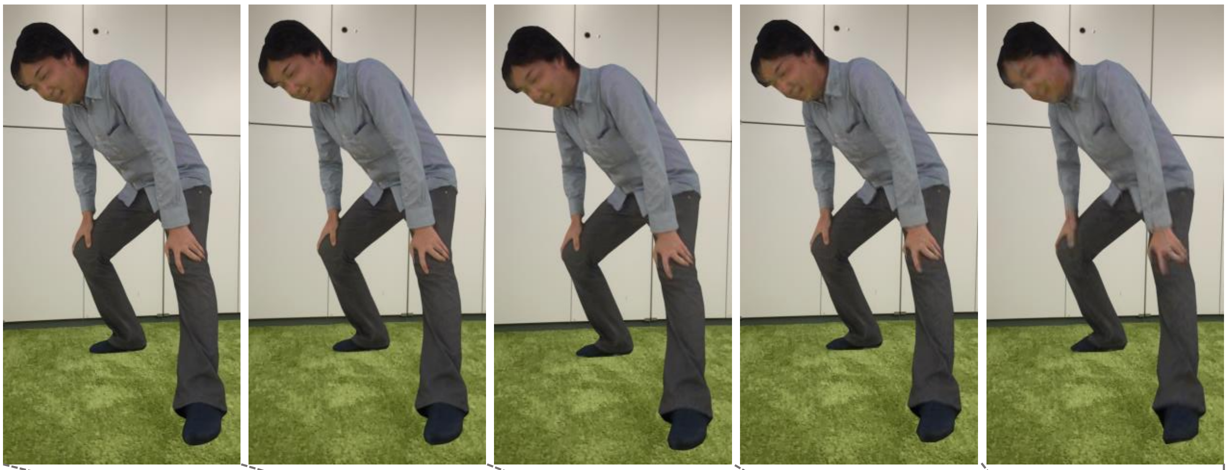
Figure 8 – Description of volumetric data

```
volumetric_data_frame {
  number_of_vertices              Uint32 (A)
  number_of_indices               Uint32 (B)
  size_of_texture_data            Uint32 (C)
  for ( i=0; i < A; i++) {
    vertex_3D_coordinates[3]      Float16 × 3 (X, Y, Z)
  }
  for ( i=0; I < A; i++) {
    vertex_UV_coordinates[2]      Float16 × 2 (U, V)
  }
  for ( i=0; i < B/3; i++) {  /* (B/3) equals the number of faces */
    vertex_indices_set[3]         Uint16 × 3 (V1, V2, V3)
  }
  for ( i=0; i < C; i++) {     /* texture image (C bytes)  */
    JPEG_compressed_data          Uint8
  }
}
```

Figure 9 – Structure of data frame

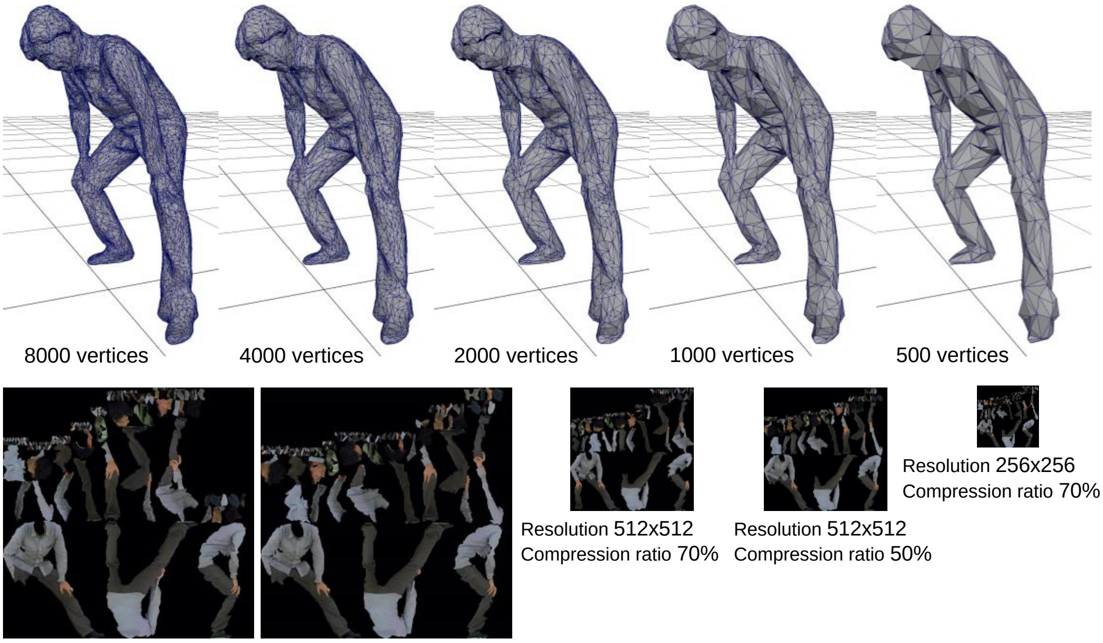
of the AR content. The signalling information contains a list of multiplexed objects in JavaScript object notation (JSON) format described in ‘IETF (5).’

In the UDP/IP packet including the head fragment of a data frame, there is an object identifier (OID) and UTC based PTS. OID is a unique value associated with a 3D object. A 3D object can be transmitted using multiple representations in different resolutions and bitrates, as



Geometry					
Number of vertices	8000	4000	2000	1000	500
Number of faces	16004	8004	4004	2004	1004
Texture					
Resolution (pixels)	1024x1024	1024x1024	512x512	512x512	256x256
Compression ratio (%)	70	50	70	50	70
Data amount					
Single frame size (Byte)	337,666	189,558	100,000	58,727	32,522
Bitrate at 30 fps (Mbps)	81.0	45.5	24.0	14.1	7.8

Figure 10 – Examples of visual qualities at different bitrates



8000 vertices

4000 vertices

2000 vertices

1000 vertices

500 vertices

Resolution 256x256
Compression ratio 70%

Resolution 512x512
Compression ratio 70%

Resolution 512x512
Compression ratio 50%

Resolution 1024x1024
Compression ratio 70%

Resolution 1024x1024
Compression ratio 50%

Figure 11 – geometries and texture images used in Figure 10

shown in Figure 10. Transmission bitrate varied from 7.8 Mbps to 81.0 Mbps, when the number of vertices in the geometry varied from 500 to 8000 and the resolution of the texture varied from 256x256 pixels to 1024x1024 pixels. Packet sequences of the multiple representations of the same 3D object have the same OID and different PIDs. The correspondence between OID, PID, and bitrate is described in the signalling information.

AR Player Application on Mobile Devices

We developed an AR player application for iOS/iPadOS that supports AR real-time playback of 3D objects. The AR player is based on the AR assistance functions provided by ARKit in iOS/iPadOS and uses the Unreal Engine 4 for real-time graphics rendering. Additionally, we implemented functions for receiving UDP/IP packets, demultiplexing sequential data frames of 3D objects, and synchronizing the AR presentation with a television. Figure 12 shows a block diagram of the receiving system.

As already noted above, a 3D object can be transmitted using multiple representations in different resolutions and bitrates. The receiver selects a representation that is suitable for its processing performance and viewport. Only packets needed for the representation are passed through by the packet level object filter. The object filter is a function close to the transport layer, which can be implemented in the demultiplex block. For example, the object filter can select appropriate representation according to the viewing distance to a 3D object. Typically, the higher bitrate representation is selected with a shorter viewing distance. Furthermore, once a 3D object is identified as being out of the viewport, the object filter discards all packets containing representations of the object.

Object Filter at Network Node

Although the object filter function can be implemented in the AR player on mobile devices, we propose the object filter be implemented at network nodes such as cloud edges and home gateways. In this case, it is more effective to reduce the processing load on the terminal device instead of implementing the object filter in the terminal device. In Figure 13, the object filter in a home gateway processes the data flows of 3D objects toward two terminal devices. Two representations are provided for each of the two objects.

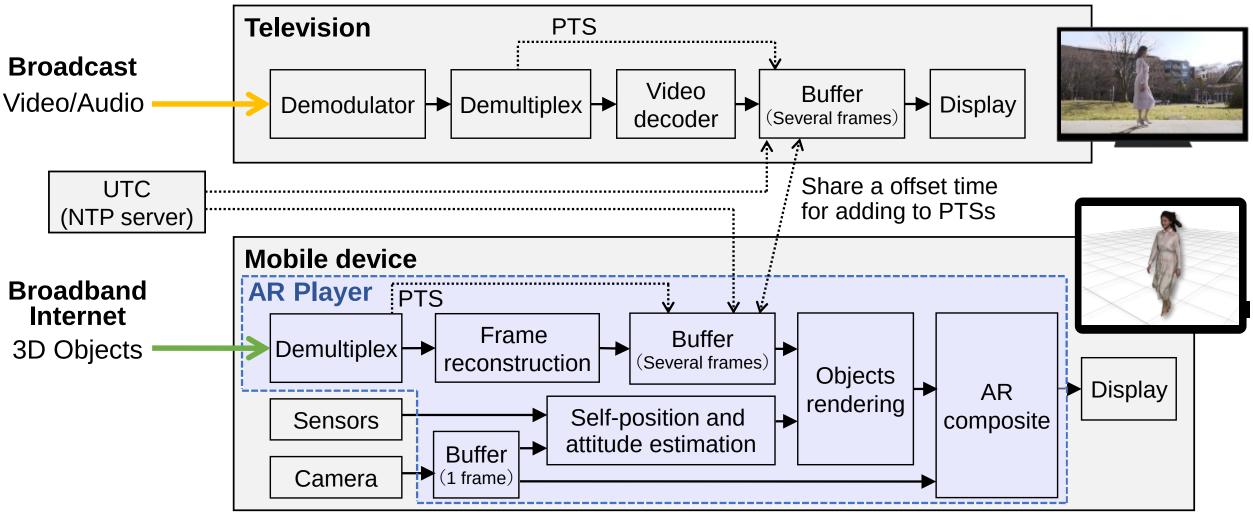


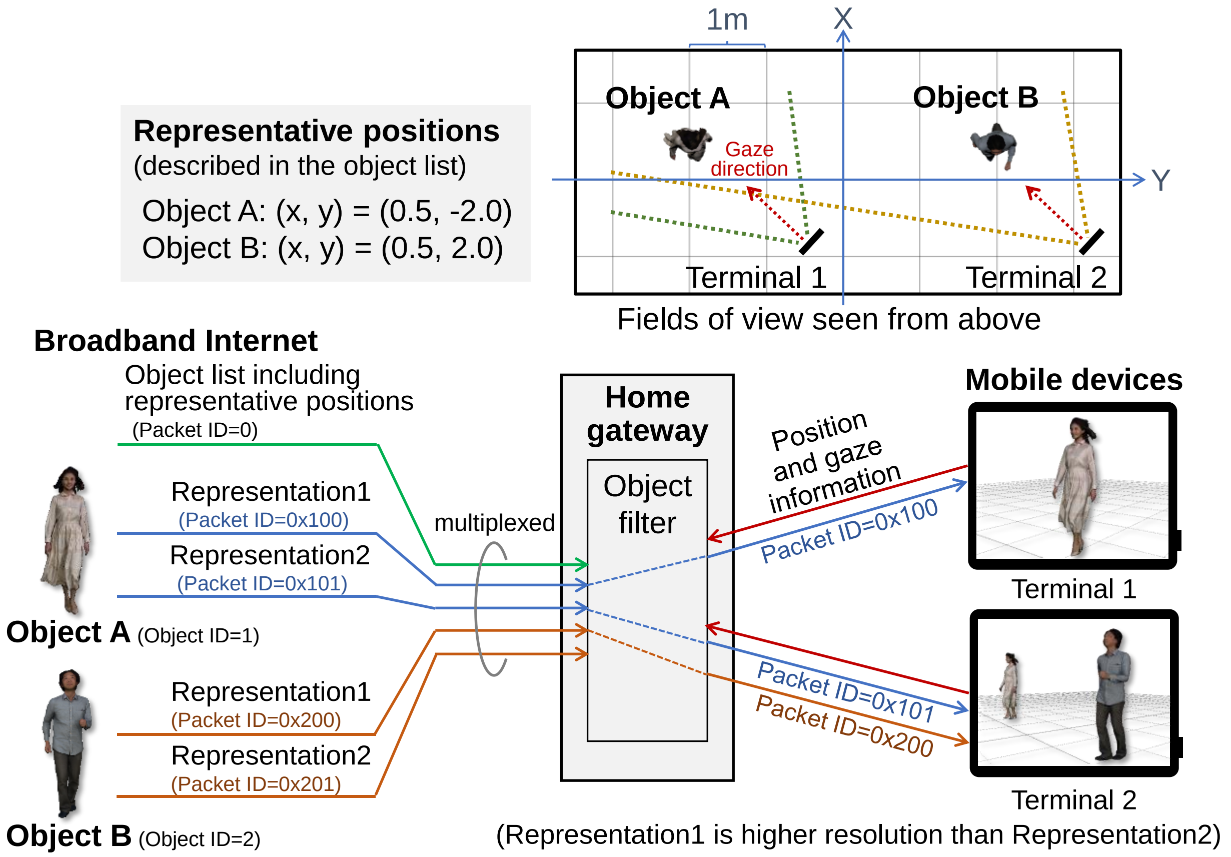
Figure 12 – Block diagram of the receiving system

The network nodes implementing the object filter need to be informed about the location of the device and the object. We added the function to output the viewing position and gaze direction to the AR player application. The viewing position is the position of the terminal device, and the gaze direction is the front side of the device camera. Hence, the viewing position and gaze direction are not the position and direction of the person holding the device, respectively. A network node with the object filter function utilizes the viewing position and gaze direction information to determine the suitable representation that the receiving device needs. Then, the network node passes only the packets the device needs. To inform the network node of the positions of the 3D objects, we included representative position information in the object list as signalling information.

FUNDAMENTAL EXPERIMENT OF OBJECT-BASED TRANSMISSION

To evaluate the object filter function described above, we conducted an experiment of object-based transmission using sequential volumetric data, running at 30 fps. Three representations of a 3D object were transmitted to a home gateway in an UDP/IP multicast. The home gateway passed through an optimal representation to a terminal device in an UDP/IP unicast over Wi-Fi (IEEE 802.11ac). The 3D object was placed in a fixed position in real space using AR, and a viewer moved with the terminal device to change the viewing distance. Figure 14 shows the appearance of the experiment at a viewing distance of 2m.

Figure 15 shows images as the viewing distance was changed from 2m to 4m. The images from (A) to (D) are untrimmed screenshots of the AR player's full viewport. The 3D object



(Representation1 is higher resolution than Representation2)

Figure 13 – Object filer in a home gateway processing data flows of 3D objects

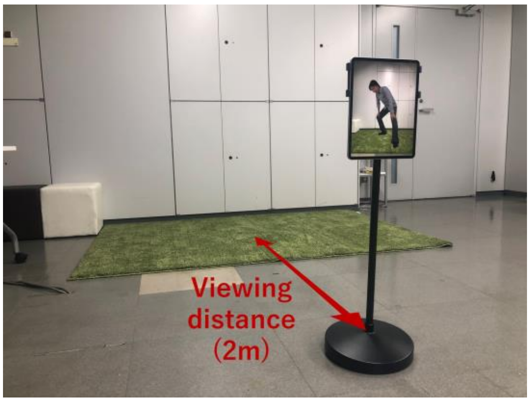
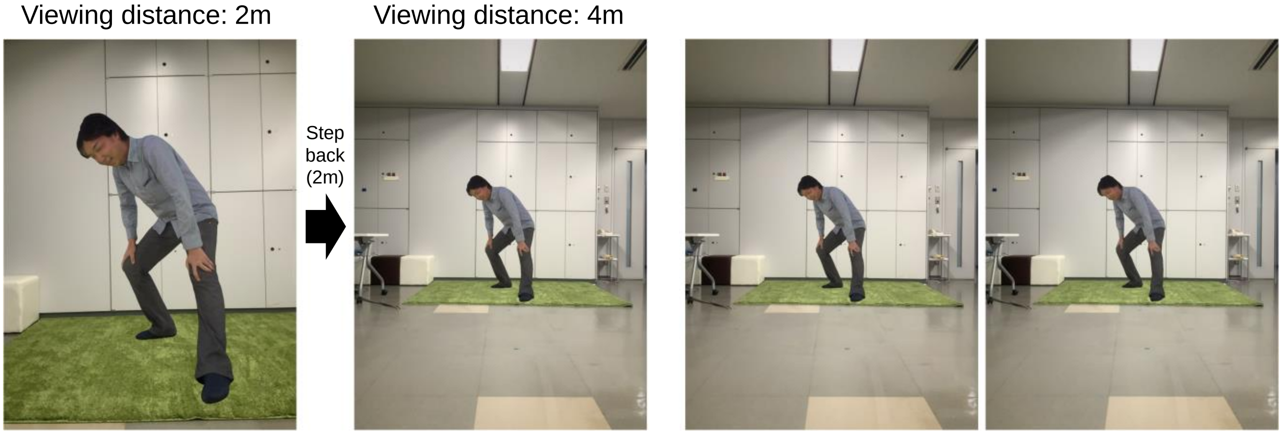


Figure 14 – Appearance of the experiment at a viewing distance of 2m



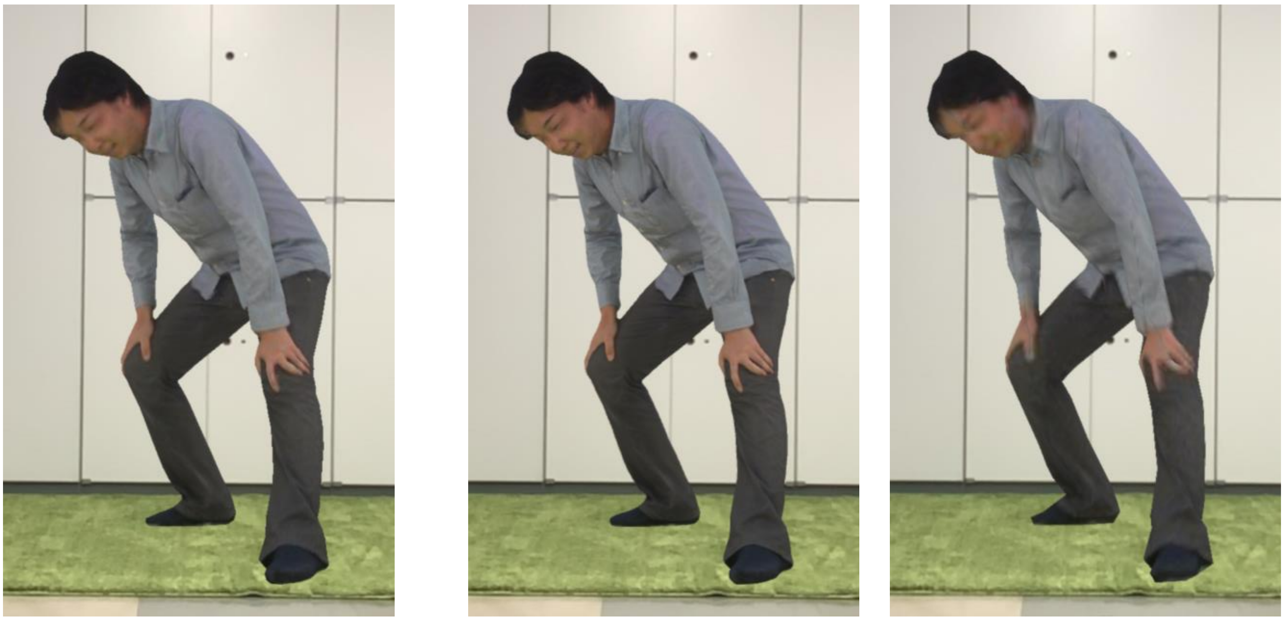
(A) Bitrate: 81.0 Mbps

(B) Bitrate: 24.0 Mbps

(C) Bitrate: 81.0 Mbps
(w/o switching representations)

(D) Bitrate: 7.8 Mbps
(switching to lower bitrate)

Figure 15 – Screenshot images of AR player (not trimmed)



(B) Bitrate: 24.0 Mbps

(C) Bitrate: 81.0 Mbps
(w/o switching representations)

(D) Bitrate: 7.8 Mbps
(switching to lower bitrate)

Figure 16 – Screenshot images of AR player (trimmed from Figure 15)

looks stationary because the same frame has been selected for easy comparison of image quality. Image (A) shows a viewport of the terminal receiving a representation at 81.0 Mbps at the viewing distance of 2m. Then the viewer stepped back to the viewing distance of 4m. Image (B) is the viewport after switching to the representation at 24.0 Mbps. Images (C) and (D) show how it would look without switching the representation for comparison and how it would look if a representation at a lower bitrate is selected, respectively. Figure 16 shows trimmed and zoomed up images of images (B), (C), and (D). Comparing image (B) and (C), there is no significant difference in appearance, although there is a substantial difference in bitrate. There is image quality degradation from (B) to (D). Nevertheless, the image quality of (D) might be sufficient if the object is not the main performer in the AR content composed of multiple 3D objects.

The above suggests the available representations, and the viewing distances to switch them should be designed according to the content properties such as the order of importance of the objects. Such additional information should be transmitted as metadata in the signalling information.

CONCLUSION

In this paper, we described enhanced viewing experiences realized by the integration of UHDTV program and AR content. Furthermore, we proposed an object-based real-time streaming system of dynamic 3D geometry and texture for AR played on mobile devices, which synchronizes with the UHDTV broadcast. In the proposed object-based transmission using multiple representations, the network node and terminal device can optimize the processing load by only filtering the necessary data at the packet level of the lower layer before the graphic rendering.

We plan to continue further research on distribution mechanisms and applications for AR content. We are going extend the proposed system to include additional features such as GL Transmission Format (glTF2.0), geometry compressions by Google Draco, supporting AR glasses, and interactive 3D sound based on the listening position. We also plan to produce contents containing multiple objects of performers and conduct transmission experiments.

REFERENCES

1. ISO/IEC 23008-1, 2014, High efficiency coding and media delivery in heterogeneous environments: MPEG media transport

2. Y. Kawamura, Y. Yamakami, H. Nagata and K. Imamura, 2019, Real-Time Streaming of Sequential Volumetric Data for Augmented Reality Synchronized with Broadcast Video, Proceedings of ICCE-Berlin, pp.280-281

3. A. Collet, M. Chuang, P. Seeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, S. Sullivan, 2015, High-quality streamable free-viewpoint video, ACM Transactions on Graphics (ToG), vol 34(4), no. 69

4. Wavefront Technologies, Appendix B1. Object Files (.obj), Advanced Visualizer Manual, http://www.cs.utah.edu/~boulos/cs3505/obj_spec.pdf

5. IETF RFC 7159, 2014, The JavaScript Object Notation (JSON) Data Interchange Format, https://tools.ietf.org/html/rfc7159