Free-Viewpoint AR Streaming Technology

Yuki Kawamura
Science & Technology Research Laboratories, Japan Broadcasting Corporation
(Currently, Media Innovation Center, General Media Administration,
Japan Broadcasting Corporation)

1. Introduction

The start of 8K broadcasting, which has evolved through quantitative improvement of specification of two-dimensional (2D) images, has led to expectations that media technology will evolve in an axis other than the improvement of specifications. Therefore, using Augmented Reality/Virtual Reality (AR/VR) technology, which has been attracting a great deal of interest since the beginning of 8K broadcasting, we propose a new viewing style that combines three-dimensional (3D) content with broadcasting, and is engaged in research and development of a transmission technology to achieve this viewing style[1]. The proposed transmission technology aims to efficiently stream 3D contents in order to support live broadcast programs in the future. For the STRL Open House 2022[2], we created content that enables experiencing the proposed new viewing style based on the NHK Special Dinosaur Super World. Viewers board a time capsule with a reporter and travel back in time to the days when dinosaurs existed. While watching the Dinosaur Super World Immersive Special Edition, viewers can enjoy 3D content, such as the Spinosaurus, Ammonite, and other dinosaurs that appear in the program, as AR content. The experience video can be viewed through the link in reference[2].

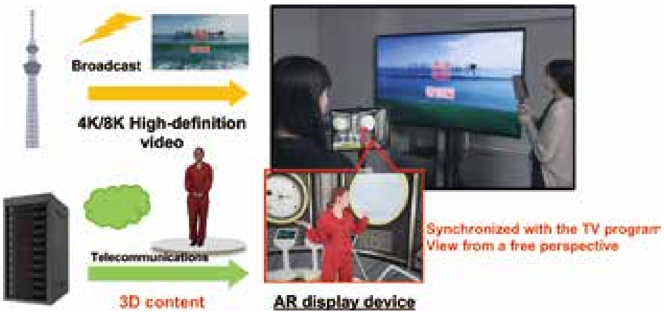
This paper first outlines and describes the requirements of the new viewing style we proposed. Next, we will discuss object-based transmission, which is being researched and developed for the efficient transmission of 3D contents, and the mechanism for improving transmission efficiency by applying object-based transmission. Finally, using the content exhibited at the STRL Open House 2022 as an example, we will introduce a concrete method for creating data to be delivered by actual object-based transmission and packaging the data as a single content.

2. Outline of the proposed viewing style and requirements

Figure 1 shows an overview of the proposed new viewing style. Three-dimensional content that is transmitted via telecommunication (Internet) and synchronized with 2D high-definition video by television broadcasting is presented to mobile display devices by AR. The TV program and 3D content share the same time axis and story, making it possible to render a new visual expression by linking the two contents. For example, in the content exhibited at the STRL Open House 2022, the reporter shown on the TV screen disappears from the TV screen and is transmitted to the AR display device as 3D content. In addition, the ability to view 3D content from a free viewpoint enables providing a new viewing experience not possible for conventional 2D images.

One of the requirements for the above-mentioned service is to stream 3D content in real time. This is to enable reducing the viewer's interaction cost related to downloading the 3D content linked to the broadcast in advance and to synchronize the content with the live broadcast program. On the other hand, since 3D content generally involves a large amount of data, it requires efficient transmission methods. To address this requirement, we are conducting R&D on the object-based transmission method[3].

■ **Figure 1: Proposed viewing style**


Broadcast
4K/8K High-definition video
Telecommunications
3D content
AR display device
Synchronized with the TV program
View from a free perspective

3. Object-based transmission

The proposed object-based transmission method is a proprietary transmission protocol based on User Datagram Protocol (UDP)/Internet Protocol (IP) that considers each performer and background content to be a single object and transmits it in an identifiable state at the packet level. Figure 2 shows the frame structure of object-based transmission. In object-based transmission, the encoded 3D data described below is framed for transmission by describing the information to uniquely identify the 3D object, called Packet ID, in the header part. Then, each object is framed, and metadata, such as Presentation Time Stamp (PTS) to be presented in the payload header, are described and multiplexed. This method, which enables flexibly processing each object at the packet level unlike when transmitting multiple 3D models together, can be applied to improve transmission efficiency. For example, a 3D model of a motionless background can be transmitted at a lower frame rate than a 3D model of a moving performer, thus reducing the volume of data equivalent to the compressed frame rate. It is also possible to adjust the total amount of transmitted data by assigning priority to each 3D model and adjusting the quality of the 3D model itself.

3.1 3D model for transmission

The 3D model for transmission assumes encoding as mesh geometry and texture image for each frame. Google Draco[4] is used for the compression of mesh geometry, JPEG is used for the compression of texture images, and GLB File Format[5], a binary format of glTF (GL Transmission Format) 2.0, is used for the transmission frame format. A moving 3D model is similar to the mechanism for continuously displaying frames of still images to make a movie. In a single file, a static 3D model is transmitted at around 30 frames per second (fps) and is expressed by continuous rendering on the display device. On the other hand, a 3D model of a static background object transmits the same still image frame at a rate of around 1 fps. Periodic transmission, even for background objects, is carried out for cases when viewers start viewing 3D content midway through.

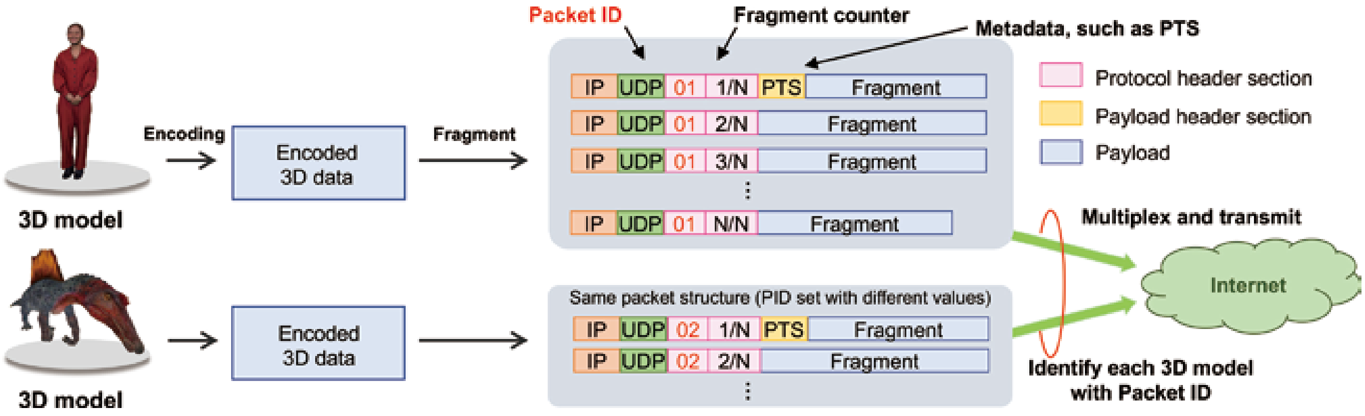
For convenience, the data of the 3D model that is actually distributed by object-based transmission is managed as sequential GLB data with the number of frames to be transmitted as the file name in a folder with each object name as the folder name. For example, the transmission data of a reporter, which is regarded as a single object to be transmitted at 30 fps in the 120-second STRL Open House content, is stored in a folder called “reporter” as sequential data labeled from “00001.glb” to “03600.glb” in which still image frames of the 3D model of the reporter are encoded in the above format.

3.2 Prototyping the viewing app

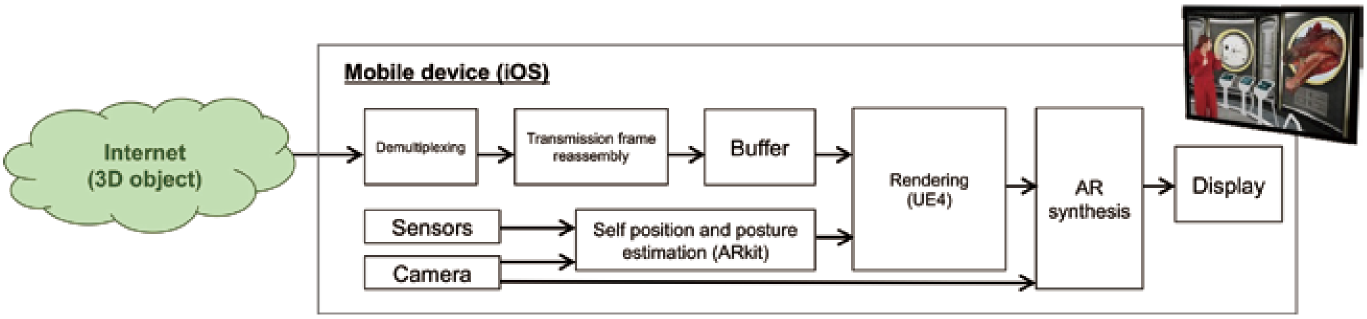
An application that receives 3D data delivered by object-based transmission and renders it in AR was implemented as an app that runs on commercially available mobile devices (iPad/Apple).

Figure 3 shows the block diagram of the application. The receiving application first receives an IP packet framed with an object-based transmission protocol and separates multiple objects that are multiplexed. Next, the transmission frame is reassembled for each object and buffered until the time of rendering according to the time stamp described in the header part of the packet. Then, 3D content is continuously rendered in AR and presented to the screen projected on the terminal. For rendering, AR Kit, an AR support function of iOS/iPad OS, and the game engine (Unreal Engine 4) are used.

■ Figure 2: Frame structure for object-based transmission



■ Figure 3: Block configuration of viewing app



4. Increasing the efficiency of data transmission

We are researching and developing a mechanism to streamline the data transmission volume by applying object-based transmission. This section describes the object filter[6] for streamlining the data delivery volume by optimizing the data delivered in accordance with the audience, and the texture thinning technique[7] for reducing the volume of the 3D data delivered.

4.1 Object filter

Figure 4 shows the system configuration diagram of object-based transmission including the object filter. As shown in Figure 4, the object filter is installed on the transmission path between the distribution server and the AR display device, and the data to be distributed is optimized in accordance with the position and gaze information of the viewer.

The data to be delivered is optimized by two methods: frustum culling and resolution pattern selection. Frustum culling is a method also used in general CG rendering and only transmits objects within the field of view of the display device. Resolution pattern selection focuses on the fact that the 3D content is displayed smaller when it is far from the display device and larger when it is close to the display device. It is a method for transmitting only data with a resolution corresponding to the distance for objects with multiple resolutions. This makes it possible to carry out more efficient transmission in the section between the object filter and the display device compared with always transmitting high-resolution data. These processes are implemented simply at the IP packet level by using Packet ID as an identifier.

4.2 Texture thinning

Figure 5 shows a conceptual diagram of texture thinning. This method focuses on the fact that polygon topology and texture mapping are the same between frames for non-live-action CG models produced by CG modeling and other tools, and the same texture image is used in all frames. Texture images with a large amount of information are thinned out of the GLB data frames of the 3D model.

The data for which texture image thinning has been carried out cannot be rendered as a 3D model as it is. However, by receiving the frame that includes the texture image once on the receiving side, then caching and using the texture data again, rendering can be carried out without problems on the display device.

Here we discuss the effect of texture thinning using the Spinosaurus used in the STRL Open House 2022 as an example. The data has 63479 vertices and a texture resolution of 1024 x 1024 pixels; and without texture thinning, the transmission bit rate is about 109 Mbps. On the other hand, it was confirmed that the amount of data can be reduced to about 34 Mbps by thinning the encoded data so as to include the texture only once in 30 frames.

■ Figure 4: System configuration of object filter

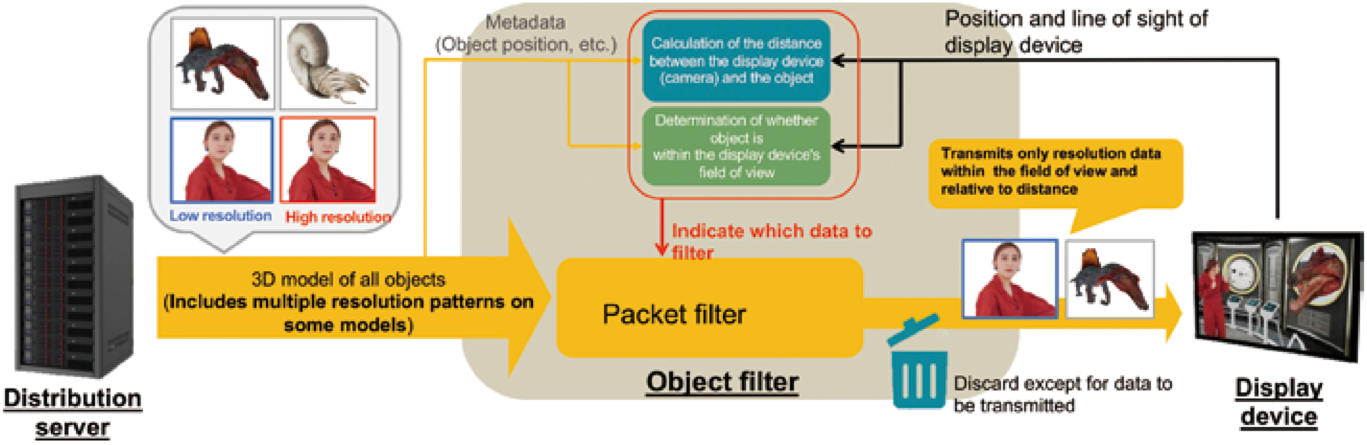
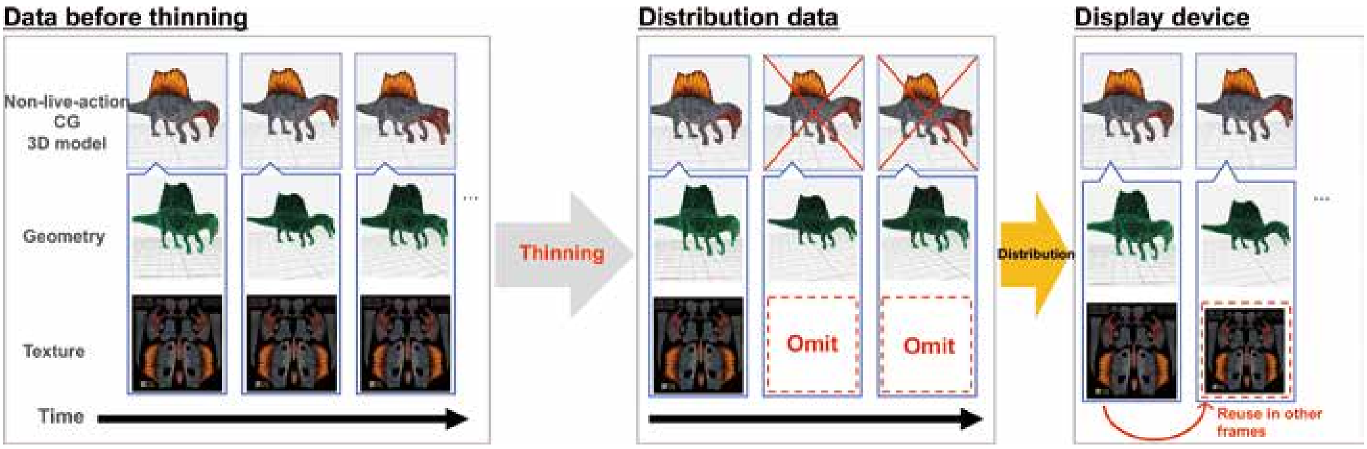

Low resolution
High resolution
Metadata
(Object position, etc.)
Calculation of the distance between the display device (camera) and the object
Determination of whether object is within the display device's field of view
Position and line of sight of display device
Transmits only resolution data within the field of view and relative to distance
Indicate which data to filter
3D model of all objects (Includes multiple resolution patterns on some models)
Packet filter
Object filter
Discard except for data to be transmitted
Distribution server
Display device

Figure 5: Texture thinning concept



5. Content packaging

In this section, we will introduce the flow for creating the data to be actually delivered by object-based transmission and packaging the data as content using the content created in STRL Open 2022 as an example.

First, a check is conducted to determine whether the content components can be decomposed into static parts and moving parts. In this example, the time capsule is broken down into two parts; namely, the floor, which is always stationary, and the wall surface and ceiling, which disappear as particles when the scene changes from land to sea, with each part treated as a separate object.

The data to be delivered is then created. Figure 6 shows the flow for creating distribution data. The figure shows a reporter as volumetric video data, the Spinosaurus as non-live-action CG data, and the time capsule (floor) as a static object. As shown in the figure, the production method can be broadly divided into three steps. First, (1) the component data with many vertices and high texture resolution is converted into sequential data in GLB format for each object, along with reducing the number of vertices and texture resolution. In this step, since the static object is the same in all frames, the data to be created consists of only one frame.

Next, (2) for each object, data is created and processed in accordance with the characteristics of the object. Here, for volumetric video, data with multiple resolution patterns is created based on object filters. For non-live-action CG objects, the texture thinning described in section 4.2. is applied. With the exception

Figure 6: Distribution data creation procedure

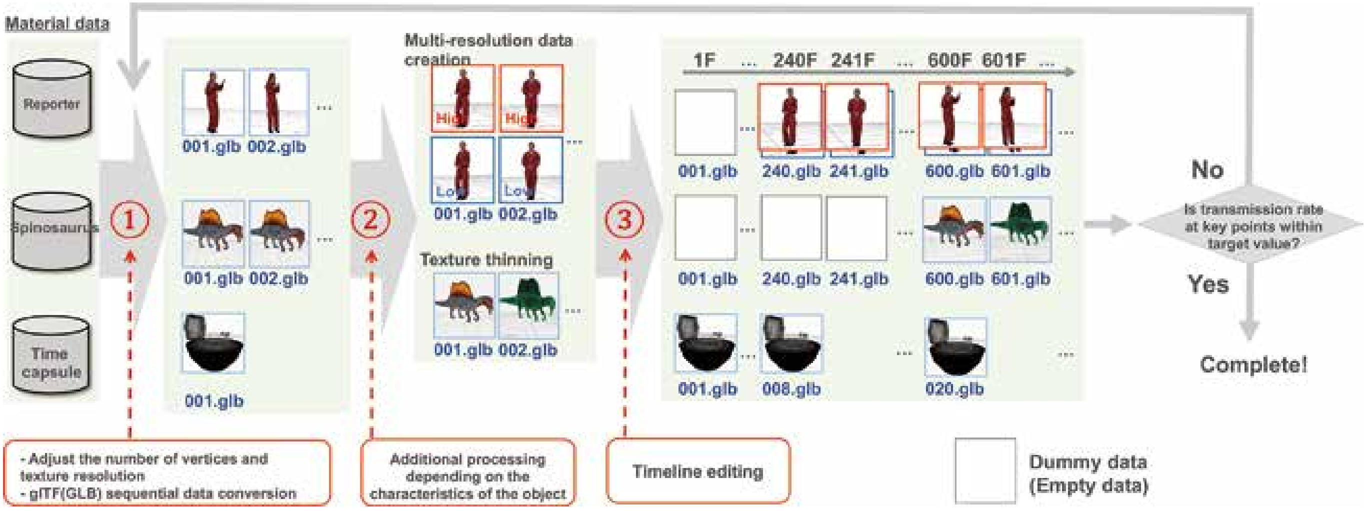
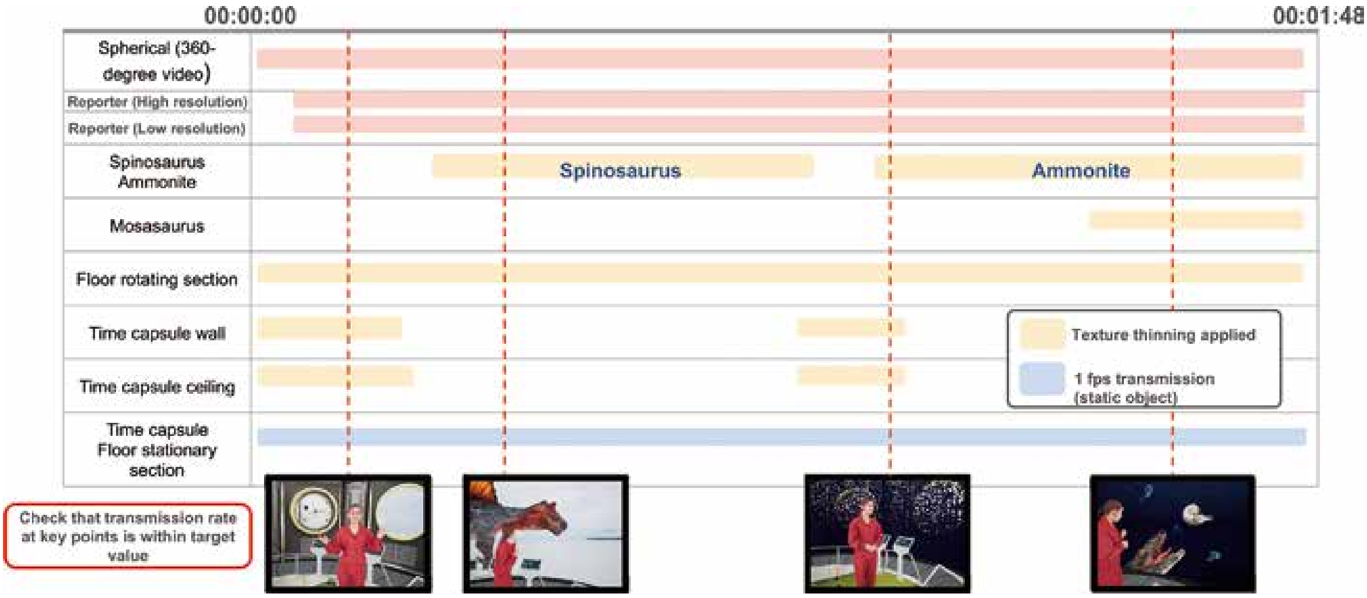

Material data
Reporter
Spinosaurus
Time capsule
①
001.glb 002.glb
001.glb 002.glb
001.glb
②
Multi-resolution data creation
High
High
Low
Low
001.glb 002.glb
Texture thinning
001.glb 002.glb
③
1F … 240F 241F … 600F 601F …
001.glb 240.glb 241.glb 600.glb 601.glb
001.glb 240.glb 241.glb 600.glb 601.glb
001.glb 008.glb 020.glb
No
Is transmission rate at key points within target value?
Yes
Complete!
- Adjust the number of vertices and texture resolution
- glTF(GLB) sequential data conversion
Additional processing depending on the characteristics of the object
Timeline editing
Dummy data (Empty data)

Figure 7: Timeline of STRL Open House 2022 content


00:00:00
00:01:48
Spherical (360-degree video)
Reporter (High resolution)
Reporter (Low resolution)
Spinosaurus Ammonite
Spinosaurus
Ammonite
Mosasaurus
Floor rotating section
Time capsule wall
Time capsule ceiling
Time capsule Floor stationary section
Texture thinning applied
1 fps transmission (static object)
Check that transmission rate at key points is within target value

of volumetric video data, each data reduction method was able to keep high-quality data with many vertices and high texture resolution at a transmission rate that enables stable streaming transmission. Thus, multiple resolutions were not prepared for non-live-action CG objects in this content.

Further, (3) timeline editing is performed to display the data of each created object at the intended timing as the actual content. This involves editing the file names. The figure shows an example of renaming the file from “00001.glb” to “00240.glb,” assuming that the first frame of the reporter’s material data is to be presented at the 240th frame of the actual content. Dummy data is inserted in frames where no object exists to turn off the display of objects.

Finally, a check is conducted to determine whether the transmission bit rate, which varies depending on the transmitted object, is within the target bit rate for stable streaming at key points.

Figure 7 shows the timeline of the content exhibited at the STRL Open House 2022.

As shown in the figure, a check is conducted to determine whether transmission is within the target bit rate, especially when multiple objects are transmitted at the same time. If not, step (1) is repeated by adjusting the number of vertices and texture resolution of the 3D model and creating the data again. If the transmission is within the target rate, actual transmission is carried out and packaging of the delivery data is completed after confirming that stable streaming transmission can be performed.

6. Conclusion

Here, we outlined the service aimed at providing a new viewing experience by synchronizing 3D contents with broadcast programs and discussed the transmission technology we are researching and developing to implement the service. We then explained the actual distribution data creation method using the contents of STRL Open 2022 as an example. Going forward, to actualize the service, we plan to develop methods to reduce the cost and automate the creation of distribution data, develop Transmission Control Protocol (TCP)/IP-based transmission technologies that enable the use of CDN (Content Delivery Network) for large-scale distribution, and develop and verify browser-based applications with low viewer interaction costs.

References

[1] Yuki Kawamura, “Development of a real-time transmission technology for free-viewpoint AR content synchronized with TV images,” Journal of the Institute of Image Information and Television Engineers, vol. 75, no.1, p.131-138, Jan. 2021 (in Japanese).

[2] NHK, “STRL Open House 2022 Exhibit 5 Free Viewpoint AR Streaming Technology,” https://www.nhk.or.jp/strl/open2022/tenji/5/ index.html.

[3] Yuki Kawamura, Yasutaka Maeda, Kensuke Hisatomi, Koichiro Imamura, “Object-based transmission of 3D content and adaptive viewing using multiple presentation methods,” IEICE Technical Report, vol.121, no. 73, SIS2021-6, p.32-36, June 2021 (in Japanese).

[4] Frank Galligan, “Draco Bitstream Specification,” https://google.github.io/draco/spec/, Oct.2017.

[5] Khronos Group, “The glTF 2.0 Specification,” https:// g ithub.com/ K hronosGroup/glTF/ blob/ma in / specification/2.0/.

[6] Yuki Kawamura, Nobuhiro Hiruma, Koichiro Imamura, “An Implementation of Visible Object Filter Gateway for Effective Streaming of Free- Viewpoint AR Content,” ITE Technical Report, vol. 45, no. 35, p. 35-40, Nov. 2021 (in Japanese).

[7] Nobuhiro Hiruma, Yuki Kawamura, Koichiro Imamura, “An efficient streaming method for 3D spatial content,” ITE Winter Annual Convention, 22B-1, 2021.