Web360^2: An Interactive Web Application for viewing 3D Audio-visual Contents

Shin Kato
The University of Tokyo
shin@hongo.wide.ad.jp

Tomohiro Ikeda
Paronym Inc.
t.ikeda.012@gmail.com

Mitsuaki Kawamorita
Bitlet LLC.
mkawamorita@bitlet.co

Manabu Tsukada
The University of Tokyo
tsukada@hongo.wide.ad.jp

Hiroshi Esaki
The University of Tokyo
hiroshi@wide.ad.jp

ABSTRACT

The use of video streaming services is expanding, and currently accounts for the majority of downstream Internet traffic. With the availability of virtual reality (VR) services and 360-degree cameras for consumer use, 3D services are also gaining in popularity. In recent years, the technology supporting for 3D representation on the Web has advanced. Users can easily utilize this technology without installing dedicated applications. In this study, we design and implement a Web application, called "Web360^2," which plays 360-degree video and object-based 3D sounds interactively on the Web. We also evaluated Web360^2 through a questionnaire survey.

1. INTRODUCTION

In recent years, the use of video streaming services has expanded rapidly. According to the 2018 Global Internet Phenomena Report released by Sandvine [1], video streaming accounts for 57.69% of all downstream Internet traffic. Further, Sandvine's 2019 Mobile Internet Phenomena Report [2] stated that YouTube accounted for 37.04% of worldwide mobile traffic, is far more than any other application. Facebook Video (2.53%) and Netflix (2.44%) were also included in the top 10 generators of application traffic. With the continuing adoption of virtual reality (VR) services, 360-degree video streaming has attracted increasing interest. YouTube, Facebook, and Netflix offer support for 360-degree video and 3D sound.

Almost all video streaming services have applications for Web browsers in addition to dedicated applications. The Web is an environment that anyone can easily use, and now supports the playing of 360-degree video and 3D sound because of recent advancements in 3D representation methods. However, there are few Web applications that can play 3D content interactively. In this study, we designed and implemented a Web application, called "Web360^2", which plays 360-degree video and object-based 3D sounds recorded during an orchestra concert and a jazz session.

In Section 2, we introduce the methods for representing 3D sounds and discuss previous interactive 3D audio-visual systems. Section 3 describes the purpose of this study. Section 4 describes the recording session and the resulting data set. Section 5 details the design and implementation of Web360^2, our interactive 3D audio-visual Web application. In Section 6, we report the results of a questionnaire on Web360^2, which was evaluated by 93 people. Finally, Section 7 concludes this paper and discusses future work.

2. RELATED WORK

2.1 3D Sound Representation

Various methods exist for representing 3D sound. MPEG-H, standardized by the Moving Picture Experts Group (MPEG) that is a working group of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), provides a 3D audio standard known as MPEG-H Part 3 [3]. MPEG-H 3D Audio supports channel-based, object-based, and higher-order ambisonics (HOA)-based audio signals [4].

2.1.1 Channel-based audio

Channel-based audio ranges from traditional 2.0-channel stereo sound to multi-channel surround sound represented by 5.1-channel and immersive sound with additional channels. It provides optimal sound performance for a predetermined number of speakers and speaker layout. However, it lacks flexibility and interactivity. Changing the playback environment, for example, increasing or decreasing the number of speakers or re-arranging the speakers, requires the use of additional sound data or modified sound data that is optimized for the new environment [5]. It is also difficult to represent sounds consistently as the listener moves within the playback environment.

2.1.2 Object-based audio

Object-based audio uses sound objects containing metadata of each 3D position in space to describe a spatial sound scene. It renders optimal 3D sound for speakers in real time using the metadata of each object. It also can play while responding to temporal changes in their coordinates. Therefore, object-based audio can generate track-

ing sound by modifying the position metadata based on the relative movement of the listener in terms of position or head rotation. In content production studios, spatial position metadata is often assigned to object-based audio by panning tools. Currently, most audio is mixed as conventional channel-based audio, not object-based audio [5]. However, object-based audio is used (sometimes together with channel-based) by Dolby Atmos [6] [7], DTS:X [8], and AuroMax [9], which are increasingly being utilized in movie theaters and home theaters.

2.1.3 Higher-order ambisonics (HOA) based audio

Higher-order ambisonics (HOA) based audio records a spherical sound scene spread around one point. In this method, the sound field is represented mathematically using spherical harmonic expansion. Unlike conventional ambisonics, which focuses only on the 0th and 1st order coefficients, HOA also focuses on the 2nd and higher orders. Using these higher orders, HOA can render higher quality sound compared with ambisonics [10]. Unlike object-based, HOA/ambisonics does not have clear spatial information, and thus it is difficult to access each sound object in the sound field [5].

In HOA/ambisonics, a special microphone called an ambisonic microphone is required to record omnidirectional sound. The resulting sound data represents a spherical sound field centered on the microphone, and sounds can be easily rotated in synchronization with the listener's head rotation. However, it is difficult to follow positional movement because the listener must always be located at the center of the spherical sound scene. Because HOA/ambisonics and 360-degree cameras share similar characteristics, HOA/ambisonics is compatible with 360-degree video. Consumer omnidirectional cameras such as the Ricoh Theta Z1 and Theta V [11] are now being equipped with ambisonic microphones, and 360-degree video services such as YouTube and Facebook support HOA/ambisonics.

2.2 Interactive 3D Audio-visual Service

Inside Music [12], the experimental Web site published in 2017 by Google Creative Lab and Song Exploder, uses WebVR and Web Audio and allows users to experience 3D sound with object-based audio in a virtual space on the Web. It is not necessary to install a dedicated application — only a Web browser is needed to employ it from any compatible device. In Inside Music, only object-based sounds, not 360-degree videos, are mapped in a virtual space. The user can move freely on a horizontal surface and turn ON/OFF each sound by clicking. The source code, which is available on Github [1], can be used for creating interactive VR applications in accordance with the Apache License, Version 2.0 [13].

In 2014, we established the Software Defined Media (SDM) consortium [14] [2] to target new research areas and markets involving object-based digital media and Internet-by-design audio-visual environments. In our previous research, we designed and implemented "SDM360^2," an interactive 3D audio-visual service with a free-view-listen point for tablet devices [15], and "LiVRation," an interactive 3D audio-visual VR system for use on head-mounted displays (HMDs) [16]. Both systems were developed using the Unity platform. A 360-degree video and object-based sounds are mapped in a virtual space and follow the viewer's head movement. The viewer can also move in virtual spaces and turn ON/OFF, extract, or emphasize each sound source, because they are separate sound objects.

[1] https://github.com/googlecreativelab/inside-music

[2] https://sdm.wide.ad.jp/

3. PURPOSE OF STUDY

The purpose of this study is to create a system that can interactively play free-viewpoint 3D content on a Web browser. Specifically, we aim to provide free-viewpoint 360-degree videos and 3D sound that can follow a user's movements and allow each sound to be controlled interactively. To achieve this objective, we created "Web360^2," a prototype application, and evaluated our prototype using a questionnaire. Because object-based sound allows each sound to be easily controlled, we adopted it instead of conventional channel-based sound or HOA-based audio, the latter of which is compatible with 360-degree video as mentioned earlier. A comparison between the related systems introduced in Section 2 and Web360^2 is summarized in Table 1. Further, we assumed the following system requirements.

Web application:
It should be easy to utilize directly from a Web browser without requiring installation of a dedicated application.

Free-view-listen
The user can freely and interactively determine the viewpoint and angle, and can automatically render and present appropriate video and sounds based on their spatial metadata.

Interactivity:
Each sound object can be switched ON/OFF by the user. High-flexibility services will be provided based on the user's interests.

Video streaming:
360-degree videos will be distributed by HTTP Live Streaming (HLS). It will be possible to reduce the waiting time before playback begins and reduce the number of unnecessary connections.

4. RECORDING AND DATASET

The media data used in Web360^2 were recorded at the Keio University Collegium Musicum concert held at Fujiwara Hiroshi Hall on the Hiyoshi Campus of Keio University on January 10, 2016, and the Musilogue Band concert held at Billboard Live Tokyo in Roppongi Midtown on January 26, 2017. Additional details on the recordings can be found in [17] and [18].

Table 1. Comparison with related systems.

System	Environment	360° video	Free view point	Free view angle	Sound control	Live streaming
SDM360^2	Unity	✓	✓	✓	✓	×
LiVRation	Unity + HMD	✓	✓	✓	✓	✓
Inside Music	Web	×	✓	✓	✓	×
Web360^2	Web	✓	✓(WIP)	✓	✓	×

4.1 Orchestra Recording

The orchestra recording session was carried out in the following manner. The recording target was a twenty-four-person Baroque orchestra featuring a number of period instruments such as the viola da gamba, théorbe, and cembalo. The recording session took place at a live concert in the 509-seat Hiroshi Fujiwara Memorial Hall at Keio University. The recorded works were all Baroque compositions, including Orchestral Suite in G Major by Johann Friedrich Fasch and two others. Fig. 1 shows the microphone layout diagram and photos of the recording session.

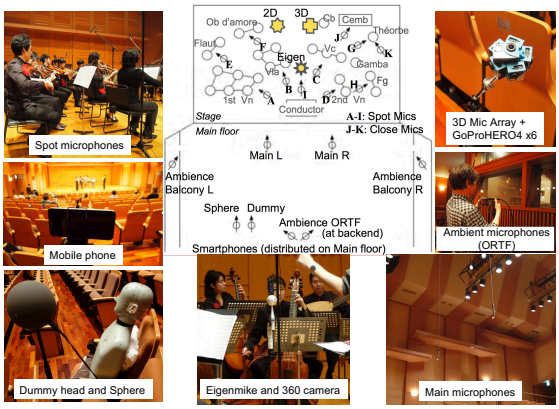

Figure 1. Camera and Microphone layout of Orchestra Recording [17]

4.2 Jazz Recording

We recorded a jazz session by the Musilogue Band at Billboard Live Tokyo in Roppongi. Billboard Live Tokyo accommodates approximately 300 people on its three floors, which include table seating (3F), sofa seating (4F), and casual seating (5F). Fig. 2 shows the formation of the band and the placement of the 360-degree camera and microphone.

The band included a drummer, electric bassist, and keyboardist (Yusuke Fujiwara, Ichiro Fujitani, and Takumi Kaneko, respectively). All of the instruments were connected to a microphones or amplifiers. The signal outputs were adjusted using a sound reinforcement (SR) mixing console and played over large speakers installed in the hall. The sound pressure level was approximately 100 dB SPL in the audience area.

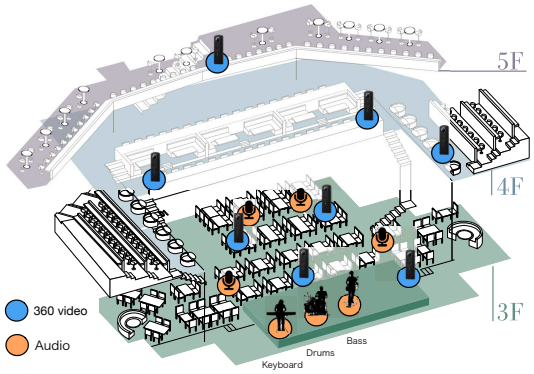

Figure 2. Camera and Microphone layout for Billboard Recording [17]

5. WEB360^2

We designed and implemented Web360^2, which fulfills the purpose and system requirements described in Section 3, and published it on Github Pages [3]. However, in the current version, there is only one viewpoint — the ability to change viewpoints is not yet supported. This section describes the system design and implementation of Web360^2.

5.1 Design

Fig. 3 shows an overview of the Web360^2 system design. In response to the user's movements and touching of audio visualizers, video and sounds are rendered in real time, which achieves system interactivity. Video and audio files are independent of each other and are played back simultaneously. We use a soundless 360-degree video and an AudioSprite [4] file that combines each sound source into one. AudioSprite arranges multiple audio files from a certain time interval and combines them into one audio file. This reduces the number of requests for playback and lessens communication overhead. Because all sound sources are arranged in one file, we do not need to consider the difference in reading delay for each sound, and it is easier to synchronize the sounds.

The 360-degree video is streamed by HLS and projected onto the inside of the user-centered virtual sphere. Each audio visualizer is rendered at a spatial coordinate inside this virtual sphere; a sound level is shown for it, and it can switched ON/OFF by touching. A tablet gyro sensor

[3] https://sdm-wg.github.io/web360square/
[4] https://github.com/tonistiigi/audiosprite

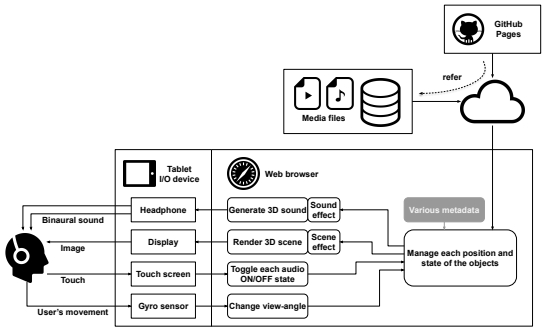

Figure 3. System design of Web360^2.

receives the user's movement as input to the system and the view angle is changed. The output sound is generated in real time based on the relationship between the current view angle and each ON state sound position/direction.

5.2 Implementation

Web360^2 was implemented using A-Frame, a WebVR framework, and the Web Audio API. Fig. 4 shows an implementation overview, and Fig. 5 shows a screenshot captured during playback. The user can toggle between playing and pausing the video and audio by tapping or clicking on the control sphere in the space (operation 1 in Fig. 5), and turn audio ON/OFF by tapping an audio visualizer (operation 2 and 3 in Fig. 5). In addition, users can change the view angle via a gyro sensor on the device (operation 4 in Fig. 5).

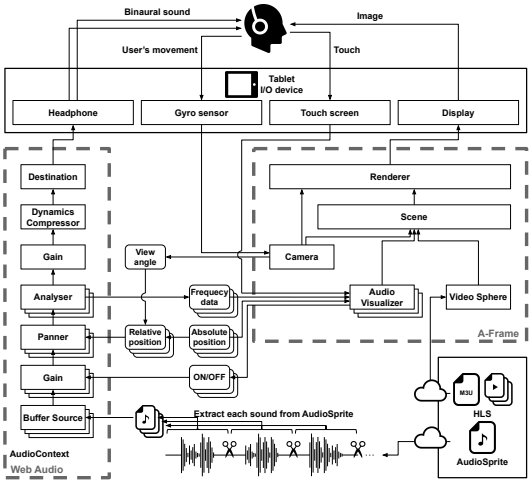

Figure 4. Implementation of Web360^2.

5.2.1 *Processes in A-Frame*

The processes in A-Frame are designed mainly to project 360-degree video on a virtual space and to map audio visualizers onto it. The 360-degree video is streamed by HLS and projected onto the inside of a virtual sphere centered on an omnidirectional camera by *a-videosphere*, an A-Frame component. Because HLS is natively supported by Safari, Edge, and some mobile device browsers, but unfortunately is unsupported by many others, we use *hls.js* [5] , a JavaScript library that implements the HLS client. The audio visualization is rendered using the frequency domain data of each sound source provided by the analyzer node of the Web Audio API.

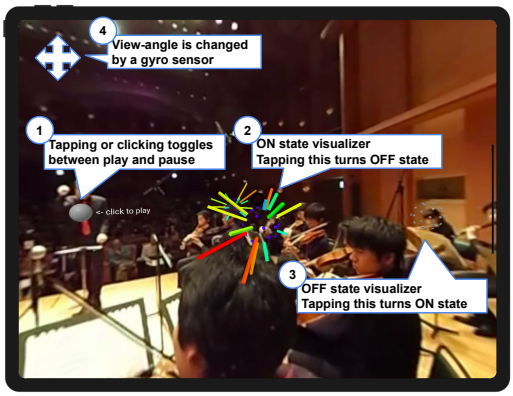

Figure 5. Screenshot of Web360^2.

In addition, some interactive user actions are input through A-Frame. After the user's movement is received by a tablet gyro sensor and inputted to the A-Frame camera object, the view angle will be changed. By touching the object rendered with A-Frame, the ON/OFF state of each sound can be changed.

5.2.2 *Web Audio API*

The Web Audio API provides the AudioContext interface and a set of functional nodes. These nodes are connected in a chain using AudioContext, and generate the output sound in real time based on the relationship between the current view angle and each ON state sound position/direction. First, each sound source is extracted from AudioSprite and inputted to each buffer source node. Next, the gain nodes express the ON/OFF state of the sound: if it is in the OFF state, the value of that node is 0. The panner nodes enable the sounds to follow the movement of the user. For example, if an audio object exists on the right-hand side of the view, the output sound is generated to be heard from the right side. Because the audio object cannot automatically recognize changes in view angle, the relative coordinates from the camera view angle and the absolute coordinates must be calculated each time, and then input to those nodes. Next, the analyzer nodes calculate the frequency domain data of each sound used to render each audio visualizer with A-Frame. The above process is performed for each sound source. Finally, each sound process is combined by the "master" gain node and the dynamic compressor node, and then output binaurally.

[5] https://github.com/video-dev/hls.js/

5.2.3 Audio visualization

The analyzer nodes of the Web Audio API can calculate time domain sound data and FFT (fast Fourier transform) based frequency domain sound data. In the audio visualization process in Web360^2, we use only frequency domain data. Frequency data is obtained by the *AnalyserNode.getByteFrequencyData()* method on the order of 60–120 times per second, using the *tick* handler of the A-Frame audio visualizer component. Fig. 6 shows the process of obtaining frequency data to render the audio visualizer. The *AnalyserNode.getByteFrequencyData()* method obtains frequency data below the Nyquist frequency. In almost all cases, there is minimal data in extremely high and low frequency bands, and the frequency band that includes all valid values is relatively narrow. We extract the upper and lower limits of the frequency in which a valid value has appeared at least once, and define this band as the valid frequency band. The valid frequency band is updated each time by the *AnalyserNode.getByteFrequencyData()* method and divided into 32 groups (in lengths as equal as possible) in the order of frequency; the average value is then calculated within each group. The audio visualizer has 32 "needles" that have one-to-one correspondence to the 32 valid frequency groups: The larger the average value, the longer and redder the needle. In detail, if the audio visualizer is in the OFF state, the needle is gray; otherwise, it changes from blue to green, to yellow, to orange, and then to red. This relationship was randomly determined because there was generally a strong correlation between close frequency groups, and because the audio visualizer acquired a clumsy shape owing to the fact that only a certain part of needles grew abnormally when we attempted to define the relationship in the order of frequency and needle position.

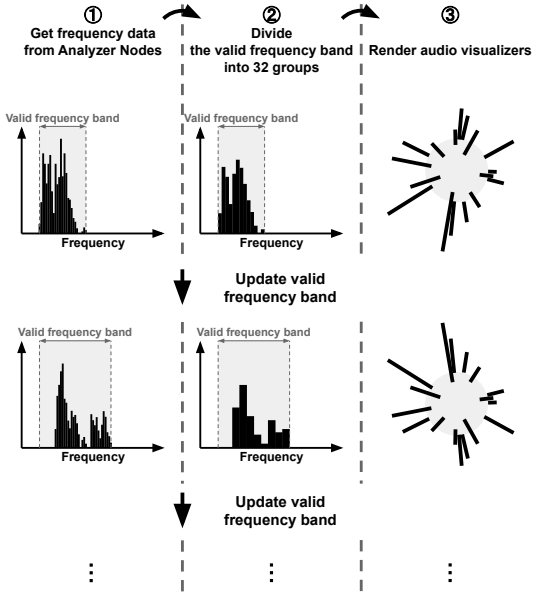

Figure 6. Audio visualizer process.

5.2.4 Synchronization between video and audio

Because the video and audio files are independent of each other, we must synchronize their playback. The playback time of the video can be obtained and set by the JavaScript property *HTMLMediaElement.currentTime*. As a similar property for audio, the Web Audio API provides the *AudioContext.currentTime* property. However, this is a read-only property and indicates the elapsed time since the AudioContext was generated, not the playback time of the audio. The value of the *AudioContext.currentTime* property continues to increase even when the audio is paused. Therefore, when the *AudioBufferSourceNode.start()* and *AudioBufferSourceNode.stop()* methods of the Web Audio API that manage the audio play/pause are called, we store the value of the *AudioContext.currentTime* property and calculate the correct playback time of the audio.

At the start of playback or the beginning of repeat playback, the video playback time is forcibly synchronized with the calculated audio playback time. Further, using the *tick* handler of the audio visualizer, the difference between the video playback time and the calculated audio playback time will be managed on the order of 60–120 times per second. Although we define a threshold value for the difference, and forcible synchronization is possible when the difference exceeds the threshold, video playback smoothness is lost with every synchronization; as a result, the QoE (Quality of Experience) is impaired. Therefore, if the video is lagging behind the audio, the video playback speed is doubled, and if the video is ahead of the audio, the video playback speed is halved. We can change the video playback speed by setting the JavaScript *HTMLMediaElement.playbackRate* property. This synchronization is "loose," and the difference between the playback time of the video and audio gradually converges within the threshold. Although modulation of the playback speed may cause some discomfort, it can help maintain smoothness. In the current version, the threshold value is set to 0.1 seconds.

6. EVALUATION

We conducted a subjective evaluation of Web360^2 through a questionnaire survey.

6.1 Evaluation Method

The subjective evaluation was carried out at the SDM consortium booth at Interop Tokyo 2019 between June 12 and June 14. According to the organizer, there were 155,801 visitors at Interop 2019, the majority of whom were employed in information systems, network engineering, sales, and research.

We asked the visitors who used Web360^2 to answer the questionnaire. In the viewing experience, two sets of headphones (Sony WH-1000XM2) were each connected via wire to two tablets (Apple iPad Pro 12.9-inch (2018), Apple iPad Pro 11-inch (2019)) that accessed Web360^2 using Wi-Fi.

First, we provided an overview of the application and explained how to use it while demonstrating its operation on

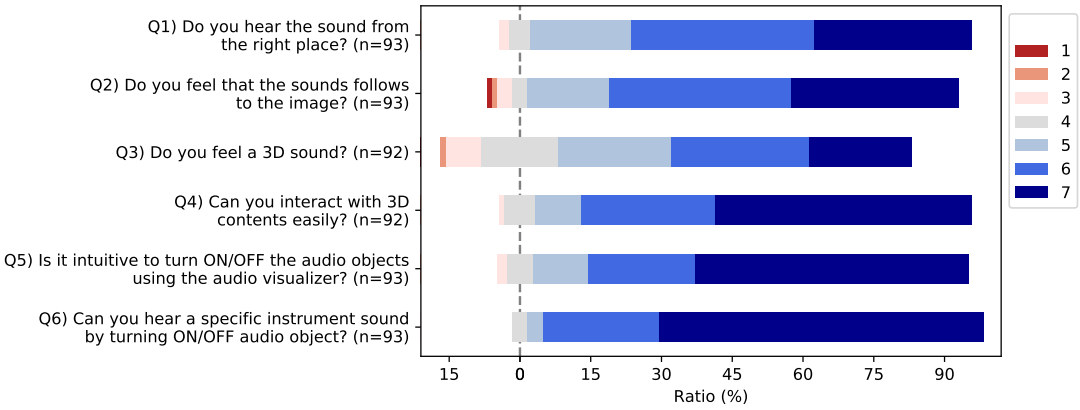

Figure 7. Response rates for questionnaires.

the Jazz recording data. We then let the visitors freely experiment with the application and evaluate it. After the experience, the visitors completed their questionnaires and we acquired the evaluation data.

6.2 Subjects

The subjects were visitors to Interop Tokyo 2019 who experimented with Web360[2]. Questionnaires were given to a total of 93 people; 81 men and nine women answered the questionnaires, while three subjects left their questionnaires unanswered. The age composition was one teenager, 24 people aged 20–29, 26 aged 30–39, 18 aged 40–49, 17 aged 50–59, and six over the age of 60; one subject did not provide their age. The subject group comprised 81 full-time employed people, three students, and five faculty members; four subjects did not provide occupation information.

6.3 Questionnaire Items

The questionnaire contained six questions (Q1–Q6), as shown below; each was evaluated using the seven-point Likert scale, ranging from 1–7 (1 = very poor, 4 = fair, and 7 = excellent).

Q1 Do you hear the sound from the right place?

Q2 Do you feel that the sounds follow the image?

Q3 Do you feel a 3D sound?

Q4 Can you interact with 3D contents easily?

Q5 Is it intuitive to turn ON/OFF the audio objects using the audio visualizer?

Q6 Can you hear a specific instrument sound by turning ON/OFF an audio object?

Q1 and Q2 asked whether the combination of the video and audio can be perceived as one unit without discomfort. In particular, Q1 queried whether there is any difference between the position of the audio visualizers on the virtual space and the perceived position/direction of the output sound. Q2 inquired about the ability of the sound to follow the movement of the view angle in response to the user's movement. Q3 inquired about the three-dimensional effect of the synthesized sound. Q4 and Q5 focused on the interactive viewing experience during the move and touch operations performed with the tablet device. Q4 queried the overall ease of operation and Q5 focused on the visualization and ON/OFF operation of the audio. Q6 queried whether the individual sound from each audio object could be heard using the ON/OFF operation. At the end of the questionnaire, we provided a text field for any additional comments or questions.

6.4 Results

Fig. 7 shows the results of the questionnaire survey. The horizontal axis represents the response rates at seven levels from 1–7; the total rate of each bar is 100%. The middle of the bar area indicating a response of 4 (fair) is aligned with the origin of the horizontal axis. While responses of 5, 6, and 7 (indicating good evaluations) are arranged in the right direction in progressively darker shades of blue, while responses of 3, 2, and 1 (indicating poor evaluations) are arranged in the left direction in progressively darker shades of red. In other words, the graph tends to be more to the right when there are many responses in the 5–7 range, and more to the left when many poor responses (3–1) are given. The vertical axis represents each question (Q1–Q6) described in the previous section 6.3; the number of valid responses is shown in parentheses at the end.

Fortunately, we were able to achieve high ratings for all the questions, and more than half of each bar indicates responses of 7 (the highest evaluation) and 6 (the second highest). In particular, for the last three questions (Q4–Q6), the bars indicate that more than half of the responses were 7's. However, in Q3, which inquired about the three-

dimensional effect of the synthesized sound, the evaluation was dispersed, and there was a relatively high percentage (8.7%) of poor evaluations (3–1). We believe that this was caused in part by the ambiguity of Q3. It is thought that multiple elements, including direction, depth, and spatial realism, form the three-dimensional effect of sound. It was difficult for the subjects to clearly understand which element the question was referring to, and we did not know which element the subjects were referring to. The stereoscopic effect of sound is an important factor in creating a high-quality 3D viewing experience. Therefore, to obtain clearer evaluation results and to improve overall performance, we will more carefully verify the precision of our questions in future surveys.

There were favorable comments regarding the intuitive operation, the ability to toggle the ON/OFF states of individual sounds, the immersive feeling of being there, and so on. In contrast, some experimenters stated that the three-dimensional effect of the sound could not be perceived easily. Since Interop is an event for information and communication technology, the participants are likely to belong to a specific subpopulation that is familiar with consuming 360-video content with tablet devices. This may have resulted in better acceptance of the proposed system. We need to clarify the implication with a larger population.

In addition, there was a request for free-viewpoint viewing in the virtual space; that feature will be implemented by using multiple 360-degree videos. There were also requests for fine volume control of each sound, and the ability to apply sound effects. We plan to add these features in future versions.

7. CONCLUSION AND FUTURE WORK

In this paper, we have presented Web360^2, a Web application that provides interactive 3D viewing. We assumed four system requirements: Web browser compatibility, free-view-listen capability, interactivity, and video streaming ability. In the design and implementation of Web360^2, the minimum requirements (other than the free-viewpoint) were satisfied. In a subjective evaluation, 93 visitors at Interop Tokyo 2019 experimented with Web360^2 and answered questionnaires. The interactive 3D viewing experience on the Web browser was evaluated highly. However, some responses identified aspects to be improved.

The SDM consortium proposes and develops "SDM Ontology [17]," a mechanism that records not only video and audio data but also detailed metadata about the recording environment; this includes location information, the orientation of the musical instruments, the music performed, venue information, and the recording process. In the future, in addition to the implementation of the free-viewpoint feature and the improvement of the synchronization between the video and audio, we will promote integration with SDM Ontology. In the current version of Web360^2, users can play only predetermined content, as described in Section 4. However, by using the SDM Ontology, content information stored on SDM Ontology can be queried and used to enhance playback. This is an advantage of Web360^2 integrating SDM Ontology; moreover, because the data stored in SDM Ontology can be converted to a virtual space easily by Web360^2, the validity of the data can be confirmed visually.

8. REFERENCES

[1] Sandvine, "Global internet phenomena report," Sandvine, Tech. Rep., 2018.

[2] ——, "Mobile internet phenomena report," Sandvine, Tech. Rep., 2019.

[3] ISO/IEC, "23008-3:2019, information technology – high efficiency coding and media delivery in heterogeneous environments – part3: 3d audio," ISO/IEC, Standard, 2019.

[4] S. Beack, J. Sung, J. Seo, and T. Lee, "Mpeg surround extension technique for mpeg-h 3d audio," *ETRI Journal*, vol. 38, no. 5, pp. 829–837, 2016.

[5] J. Herre, J. Hilpert, A. Kuntz, and J. Plogsties, "Mpeg-h 3d audio—the new standard for coding of immersive spatial audio," *IEEE Journal of selected topics in signal processing*, vol. 9, no. 5, pp. 770–779, 2015.

[6] Dolby Laboratories, "Dolby atmos®specifications," Dolby Laboratories, Isuue 3, 2015.

[7] ——, "Dolby atmos®home theater installation guidelines," Dolby Laboratories, Tech. Rep., 2018.

[8] DTS, Inc. Home theater sound gets real. DTS, Inc. [Online]. Available: https://dts.com/dtsx (accessed Feb. 26, 2020).

[9] Auro Technologies, "Auromax®next generation immersive sound system," Auro Technologies, Tech. Rep., 2015.

[10] M. Frank, F. Zotter, and A. Sontacchi, "Producing 3d audio in ambisonics," in *Audio Engineering Society Conference: 57th International Conference: The Future of Audio Entertainment Technology–Cinema, Television and the Internet*. Audio Engineering Society, 2015.

[11] Ricoh Company, Ltd. 360-degree camera ricoh theta. Ricoh Company, Ltd. [Online]. Available: https://theta360.com/ (accessed Feb. 26, 2020).

[12] Google Creative Lab and Song Exploder. Song exploder presents: Inside music. Google Creative Lab. [Online]. Available: https://experiments.withgoogle.com/webvr/inside-music/view/ (accessed Feb. 26, 2020).

[13] Apache Software Foundation. Apache license, version 2.0. Apache Software Foundation. [Online]. Available: https://www.apache.org/licenses/LICENSE-2.0 (accessed Feb. 26, 2020).

[14] M. Tsukada, K. Ogawa, M. Ikeda, T. Sone, K. Niwa, S. Saito, T. Kasuya, H. Sunahara, and H. Esaki, "Software defined media: Virtualization of audio-visual services," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–7.

[15] M. Tsukada, Y. Komohara, T. Kasuya, H. Nii, S. Takasaka, K. Ogawa, and H. Esaki, "Sdm360^2: An interactive 3d audio-visual service with a free-view-listen point," *Information Processing Society of Japan Transaction of Digital Contents Creation (DCON)*, vol. 6, no. 2, pp. 10–23, 2018, (Japanese).

[16] T. Kasuya, M. Tsukada, Y. Komohara, S. Takasaka, T. Mizuno, Y. Nomura, Y. Ueda, and H. Esaki, "LiVRation: Remote VR live platform with interactive 3D audio-visual service," in *IEEE Games Entertainment & Media Conference (IEEE GEM) 2019*, Yale University, New Haven, CT, U.S., 2019, pp. 1–7.

[17] R. Atarashi, T. Sone, Y. Komohara, M. Tsukada, T. Kasuya, H. Okumura, M. Ikeda, and H. Esaki, "The software defined media ontology for music events," in *Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music*. ACM, 2018, pp. 15–23.

[18] M. Ikeda, T. Sone, K. Niwa, S. Saito, M. Tsukada, and H. Esaki, "New recording application for software defined media," in *Audio Engineering Society Convention 141*. Audio Engineering Society, 2016.