Multi-View Video and Audio Transmission with MPEG-DASH and Its QoE

Toshiro Nunome and Hiroaki Tani
Department of Computer Science and Engineering, Graduate School of Engineering,
Nagoya Institute of Technology, Nagoya 466–8555, Japan
nunome@nitech.ac.jp, hiroaki@inl.nitech.ac.jp

Abstract—**In this paper, we assess QoE of multi-view video and audio (MVV-A) IP transmission with MPEG-DASH by a subjective experiment. An MVV-A system by means of MPEG-DASH is implemented. We then compare QoE of transmitting a selected viewpoint from four viewpoints by the user with that of transmitting a pre-determined viewpoint (i.e., single-view). We assess QoE multidimensionally with many adjective pairs. We then show that MVV-A by means of MPEG-DASH can enhance QoE under lightly loaded condition.**

Keywords*—*MVV, MPEG-DASH, HTTP, streaming, audiovisual

I. Introduction

Video streaming services over the Internet have been very popular. Most of the services currently employ Web based transmission techniques with HTTP/TCP.

In previous video streaming services over HTTP/TCP, a file has been employed for a unit of request and transfer. On the other hand, adaptive streaming, which can adaptively change quality of streaming data according to network conditions, has gained much attention recently. From about 2008, major IT vendors provide adaptive streaming services over HTTP/TCP with their dedicated techniques. Examples of the services are Smooth Streaming by Microsoft, HTTP Live Streaming (HLS) by Apple, and Dynamic HTTP Streaming by Adobe [1]. The systems do not have interoperability, and then contents and software modules are required for each service.

For unification of the method of adaptive streaming, MPEG-DASH (Dynamic Adaptive Streaming over HTTP) [2] has been standardized. Owing to this standardization, it is easy to develop systems for the services. In particular, for providers of streaming services, interoperability and re-usability are large merits. Thus, MPEG-DASH is a promising technique for growing the market [1].

As a new type multimedia service over the Internet, Multi-View Video (MVV) [3], in which users can watch video from various viewpoints, has been achieving much attention. MVV can provide higher presence to the users than the previous single-view video because the users can change the viewpoint. In current video streaming services, the users can watch only the same viewpoint given by the sender even if they move their viewpoints in front of the display. With MVV, the users can receive video specified for them. We can consider various applications of MVV such as entertainment, sports, sightseeing, and education among others.

Because the Internet basically provides a best effort service, when a communication line is congested, interruption and latency occur in downloading audio and video data. In such the case, QoS (Quality of Service) degrades, and QoE (Quality of Experience) [4] also degrades. For users, who are recipients of the service, improving QoE is important.

There have been many studies regarding MPEG-DASH. In [5], performance evaluation of adaptive transmission algorithm is carried out. For enhancement of QoE, the algorithm restricts changes of the video bitrate when the stored data size in the buffer is smaller than predetermined threshold. Reference [6] performs a subjective experiment to evaluate the effect of initial delay, pause, and variation of encoding bitrate on users' experience. However, the studies consider single-view video streaming and then do not assess QoE of MVV systems.

As for MVV on MPEG-DASH, a transmission method of MVV for RTP/RTSP (Real-time Transport Protocol/Real Time Streaming Protocol), that for HTTP Progressive Download, and that for MPEG-DASH are proposed in [7]. The paper compares the three methods on occupied data rate and viewpoint change delay; however, the paper does not evaluate QoE.

In our previous work such as [8], QoE of MVV-A (Multi-View Video and Audio), which is MVV accompanied with audio, IP transmission is assessed multidimensionally. However, these studies do not consider MVV-A systems with HTTP/TCP.

Thus, in this paper, we assess QoE of MVV-A IP transmission with MPEG-DASH by a subjective experiment. An MVV-A system by means of MPEG-DASH is implemented. We then compare QoE of transmitting a pre-determined viewpoint with that of transmitting a selected viewpoint from four viewpoints by the user.

The remainder of this paper is organized as follows. Section II introduces the MVV-A system with HTTP/TCP. Section III describes the experimental method. Section IV presents experimental results. Section V concludes this paper.

II. MVV-A system with HTTP/TCP

MVV-A is a system in which the user can watch contents from various viewpoints while he/she chooses the viewpoints arbitrarily. It provides high flexibility of the service for the user.

MPEG-DASH is a standardization of streaming techniques by means of HTTP; it is known as ISO/IEC 23009-1. In MPEG-DASH, to enable adaptive streaming transmission, the Web server stores video streams of various types of image size and encoding bitrate for each content. Each video data is divided into small chunks called *segments*. The client requests segments of appropriate bitrate to the server through HTTP, and then the server transmits the segments of requested bitrate to the client.

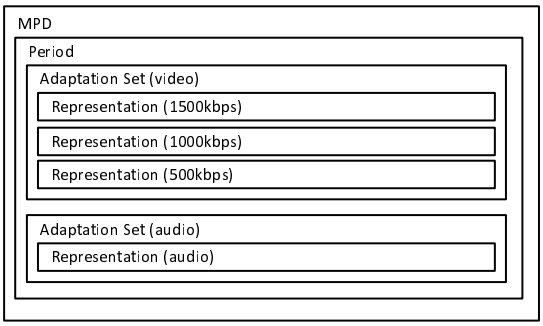

Fig. 1. MPD file for single-view video

MPD (Media Presentation Description) is employed to realize the adaptive streaming mechanism; it is described hierarchically by means of XML (eXtensible Markup Language). It depicts the structure of the content stored in the server.

The MPD file includes URL of video data, encoding method of video data, image size, encoding bitrate, encoding method of audio data, language of audio, among others. The information is described hierarchically with *Period*, *AdaptationSet*, and *Representation*. Period is a unit to compose a program or a content; it is a synchronization set of audio and video streams. AdaptationSet is a media element such as video, audio, or a caption of a language; the element is selected by the user. Representation describes the specifications of media data such as URL, video image size, audio and video bitrate.

In this study, to realize an MVV-A system with HTTP/TCP, we make an MPD file for multi-view video and audio. We then enhance a player program by means of JavaScript [9] to implement viewpoint change function.

An example of the MPD file for single-view video and audio in Fig. 1. The MPD file has two AdaptationSets for audio and video, respectively. In the AdaptationSet for video, multiple Representations exist for video encoding bitrate.

We then show an example of the MPD file for multi-view video and audio in Fig. 2. In order to create the MPD files for MVV-A, we prepare multiple MPD files for multiple viewpoints. In the basis of the MPD files for single-view video, we construct the MPD file for MVV-A; the MPD file includes AdaptationSet for each viewpoint. In addition, the AdaptationSet equips *Viewpoint* for viewpoint change function. Owing to this, we can manage multi-view video and audio of various encoding bitrate in a MPD file. The files are placed on the server as shown in Fig. 3.

Here, we explain the viewpoint change program of the MVV-A system in this paper. At first, the client requests the MPD file to the server. It receives the MPD file and a program code for viewpoint change function from the server and parses them. Next, the client requests *Cue lists*, which describes positions of segments in the media file by means of Representation. With the information, the client requests the audio and video segments for initial viewpoint to the server. When the user issues a viewpoint change request, the client decides which segments to be received and then requests them to the server.

Fig. 4 presents a screen-shot of the media player through

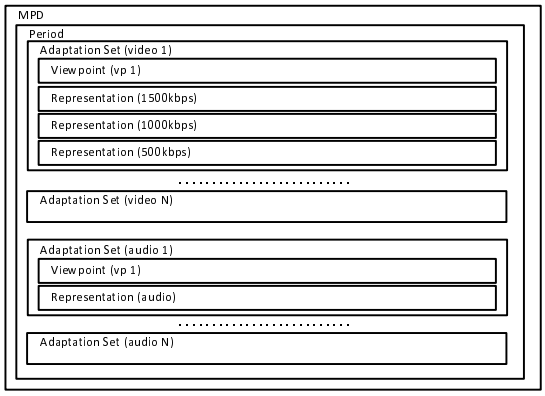

Fig. 2. MPD file for multi-view video

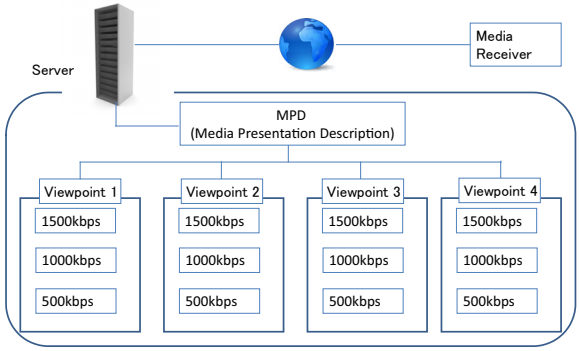

Fig. 3. Placement of files on the server

the Web browser. When the user wants to change the viewpoint, he/she pushes one of the buttons below the media player.

III. Experimental Method

A. Network configuration

Fig. 5 shows the configuration of the experimental system. The system consists of Media Server, Media Receiver, Web Server, Web Client, and two Routers. The OS of Media Server

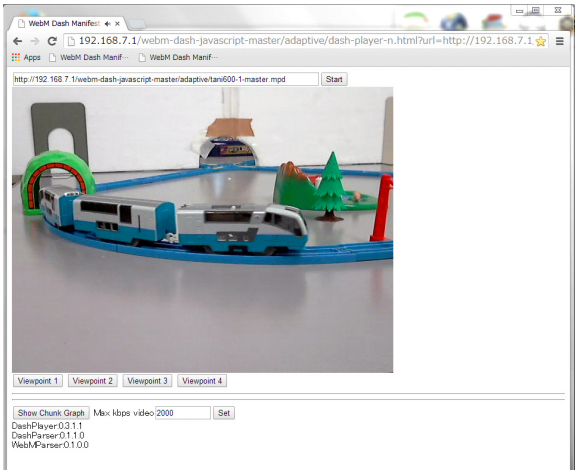

Fig. 4. Display image

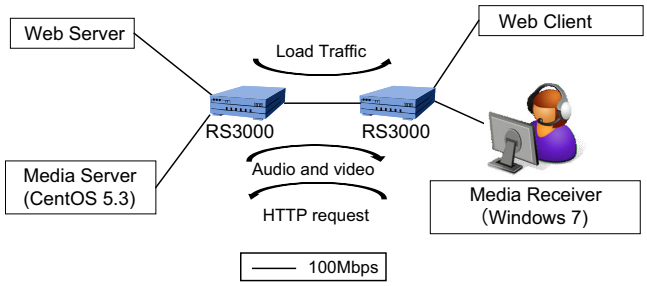

Fig. 5. Experimental network

is CentOS 5.3, and that of Media Receiver is Windows 7. The two Routers are Riverstone's RS3000. All the links are 100 Mbps duplex Ethernet.

Media Server sends the audio and video of a viewpoint to Media Receiver. Media Receiver receives these packets and outputs the audio and video decoded from them. Web Server sends the load traffic with HTTP/TCP according to requests generated by *WebStone 2.5* [10], which is a Web server benchmark tool, to Web Client. WebStone creates load on Web Server by simulating the activity of multiple clients, which are called Web client processes and which can be thought of as users, Web browsers, or other software that retrieves files from Web Server. In order to create various network conditions, we set the five patterns of the number of Web client processes to Webstone: 10, 20, 30, 40 and 50. Both Web Server and Media Server are Apache2.2 [11].

We employ Google Chrome as the Web browser for playing the audio and video in Media Receiver. To exploit the MPEG-DASH framework, we use WebM as a container format of the audio and video streams. Then, the video encoding format is VP8, and the audio encoding format is Vorbis. For creating the video stream, we exploit ffmpeg, libwebm [12], and webm-tools [13]. For generating audio and video files with the WebM format, ffmpeg supports libvpx [14] and libvorbis [15]. In order to realize adaptive bitrate streaming, libwebm arranges WebM files with sample_muxer. We utilize webm-tools for generating MPD files. We employ webm-dash-javascript [9] as a video player object. It controls video by the video element in HTML5.

The specifications of audio and video are shown in Table I. In this study, we encoded the video into three types: 500 kbps, 1000 kbps and 1500 kbps. Depending on the load condition of the network, the adaptive bitrate streaming can change the bitrate of segments seamlessly. In addition, we set the minimum buffering time (minBufferTime) to one second in the MPD file.

B. QoE assessment method

In the experiment, we compare MVV-A and SVV-A. In MVV-A, we can select a viewpoint from all the four cameras. In this study, the audio is also changed according to the viewpoint, i.e., MVV-SA in [18]. On the other hand, in SVV-A, the viewpoint is fixed; i.e., we cannot change it. In SVV-A, the users can watch the video and audio of Camera 1 only. In this paper, the audio and video are recorded in advance.

The camera arrangement is shown in Fig. 6. In the subjective experiment, an assessor watches a toy train which runs on

TABLE I. SPECIFICATIONS OF AUDIO AND VIDEO

media	item	value
audio	codec	Vorbis
	frame rate [fps]	25
	bit rate [kbps]	32
	channel	mono
	sampling rate [kHz]	8
video	codec	VP8
	frame rate [fps]	29.97
	GOP length	15
	bit rate [kbps]	500,1000,1500
	image size	640 × 480
audio and video	container format	WebM
	duration [s]	630

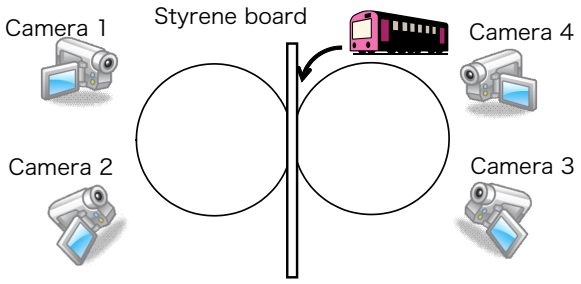

Fig. 6. Camera position

plastic rails with changing the viewpoint.

In this study, we assess QoE multidimensionally. Table II shows adjective pairs for evaluating each stimulus. The adjectives are classified into six categories: video, audio, synchronization, response, psychology, and overall quality. Abbreviated names from v1 to o1 are attached to the pairs of polar terms.

In each criterion, the assessors evaluate with the rating scale method [16]. The rating scale provides a numerical indication of the perceived quality and is expressed as a single number in the range 1 to 5. The worst grade (score 1) means the negative adjective (the left-hand side one in each pair) while the best grade (score 5) represents the positive adjective (the right-hand side one). The middle grade (score 3) is neutral. For example, each grade is defined for "v1: video is rough - smooth" as shown in Table III. Finally, we calculate the mean opinion score (MOS), which is average of the rating scale scores for all the users.

We have totally 10 stimuli to be evaluated because of the two methods (MVV-A and SVV-A) and the five patterns of the number of Web client processes. The duration of each experimental run is 20 seconds. The assessor is 20 male students in their twenties. After each experimental run, the assessor evaluates quality represented by the adjective pairs. The total time for the experiment to a subject is about 15 minutes. In order to familiar with the experiment, before the experiment, the subject practices the evaluation without load traffic.

IV. EXPERIMENTAL RESULT

A. End-to-end-level QoS

In this experiment, we installed Wireshark [17] to Media Receiver in order to measure the bandwidth consumption.

TABLE II. ADJECTIVE PAIRS FOR QOE ASSESSMENT

category	adjective pairs
video	v1: rough - smooth v2: blurred - sharp v3: difficult to grasp - easy to grasp
audio	a1: artificial - natural
synchronization	s1: out of synchronization - in synchronization
response	r1: slow - fast
psychology	p1: restricted - free p2: difficult - easy
overall	o1: bad - excellent

TABLE III. GRADES IN "V1: VIDEO IS ROUGH - SMOOTH"

score	grade
5	smooth
4	a little smooth
3	moderate
2	a little rough
1	rough

We captured all packets which were sent from Media Server to Media Receiver by Wireshark. We then counted the TCP payload size of packets received on Media Receiver.

In Fig. 7, as the end-to-end throughput, we show the average of the total amount of TCP payload size received in each experimental run, i.e., 20 seconds. The abscissa and the ordinate show the number of Web client processes and the average amount of transmitted data from Media Server, respectively.

We notice in Fig. 7 that for both SVV-A and MVV-A, the average amount of transmitted data decreases as the number of Web client processes increases. When the number of Web client processes is 10, 20, 40 and 50, the average amount of transmitted data in MVV-A is slightly smaller than that in SVV-A. This is because in MVV-A, when the viewpoint change request occurs, the server transmits the video stream of the lowest encoded bitrate, i.e., 500 kbps. In adaptive bitrate streaming with HTTP, it is common that the server transmits the lowest bitrate video in the start of transmission; this study also employs the strategy. On the other hand, in SVV-A, there is no viewpoint change request, and the server chooses to transmit video of higher encoding bitrate when the network condition is good.

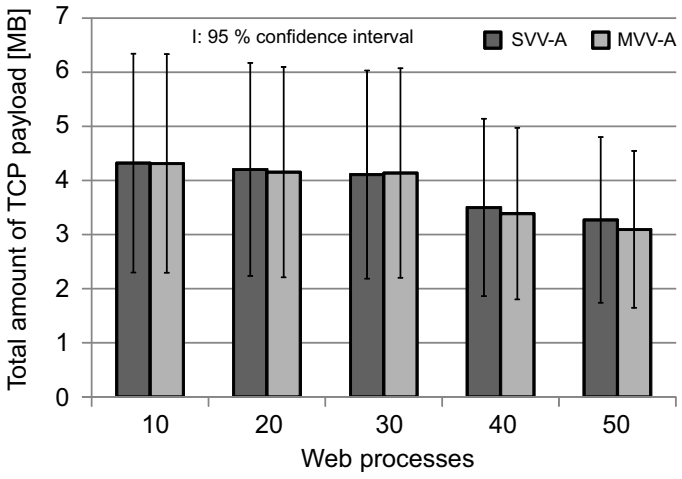

Fig. 7. Average amount of transmitted data

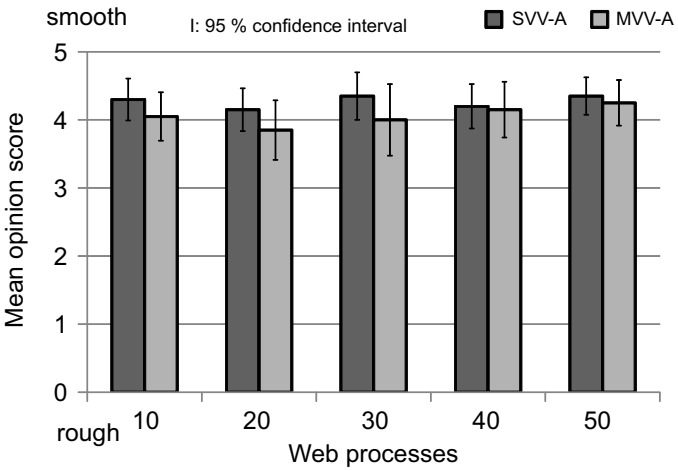

Fig. 8. MOS of "v1: video is rough - smooth"

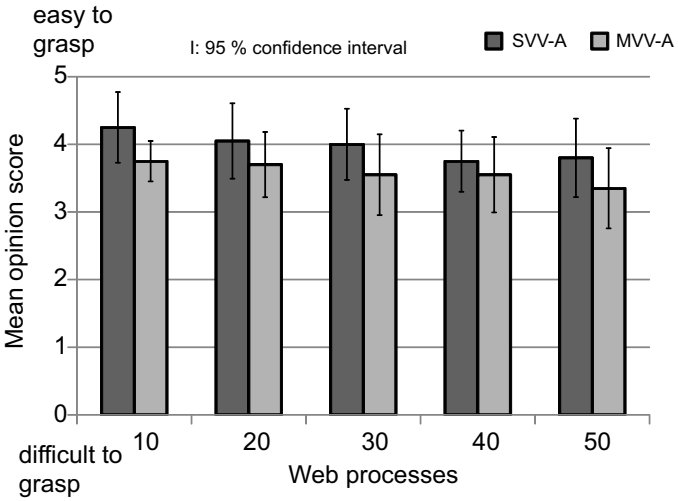

Fig. 9. MOS of "v3: video is difficult to grasp - easy to grasp"

B. QoE assessment result

In this paper, we pickup the four adjective pairs: "v1: video is rough - smooth", "v3: video is difficult to grasp - easy to grasp", "p1: restricted - free", and "o1: bad - excellent". The results are shown in Figs. 8 through 11. The abscissa and ordinate mean the number of Web client processes and MOS, respectively.

Fig. 8 shows the MOS of "v1: video is rough - smooth". We notice that for all the number of Web client processes considered here, SVV-A has larger MOS values than MVV-A. When the viewpoint change request occurs, the client starts to get data for the new viewpoint. The client needs to re-buffer the data for the new viewpoint, and the output can be awkward.

Fig. 9 depicts the MOS of "v3: video is difficult to grasp - easy to grasp". We also see that SVV-A has larger MOS values than MVV-A. This is because when the viewpoint change occurs in MVV-A, the server transmits video with the lowest encoding bitrate. Thus, the picture quality can degrade. On

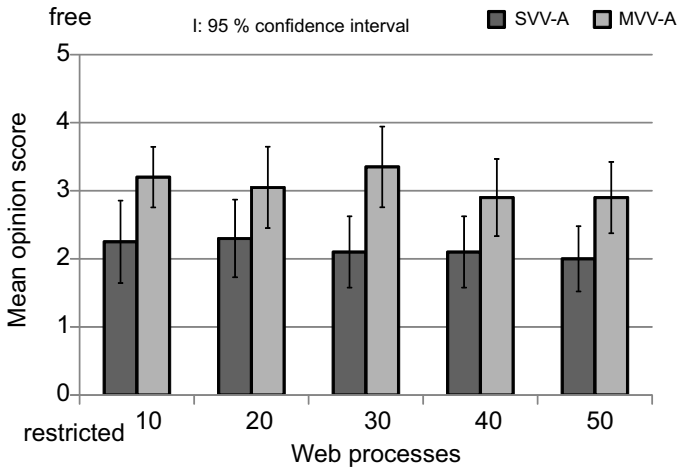

Fig. 10. MOS of "p1: restricted - free"

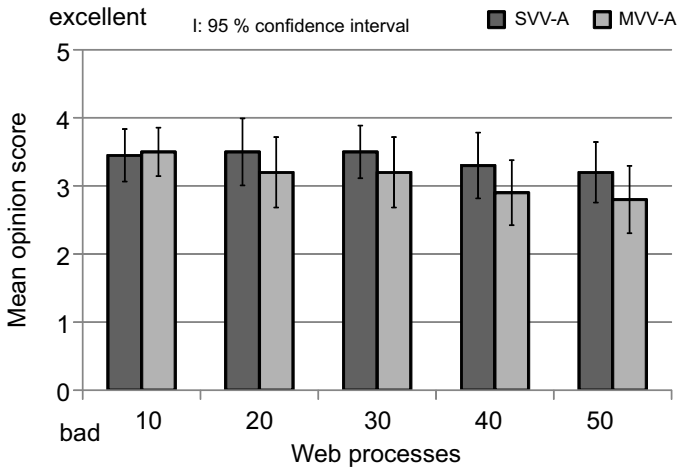

Fig. 11. MOS of "o1: bad - excellent"

the other hand, in SVV-A, the client continues to receive the same video data. Thus, SVV-A can provide continuous output of high quality video.

In Fig. 10, we find that the MOS of "p1: restricted - free" in MVV-A is larger than SVV-A for all the number of Web client processes considered here. This is because the ability of viewpoint change enhance users' feelings of freedom.

We notice in Fig. 11 that for the number of Web client processes 10, MVV-A has slightly larger MOS of "o1: bad - excellent" than SVV-A, while it has lower MOS when the number of Web client processes is larger than 10.

As we find in the above discussions, although the MVV-A system in this study tends to have lower video quality, it has high feelings of freedom, and then QoE becomes higher under lightly loaded condition. In this paper, we use the toy train as the content. In the case of seeing moving object, the ability of viewpoint change enhances users' satisfaction. However, the bad effects of the MVV-A system are dominant for overall satisfaction in the experiment. Thus, we need to devise the techniques for reducing the effect of re-buffering and picture quality degradation in viewpoint change requests.

V. Conclusions

In this paper, we implemented an MVV-A system by means of MPEG-DASH. We then compared QoE of transmitting a viewpoint from four viewpoints selected by the user with that of transmitting a pre-determined viewpoint quantitatively. As a result, we find that MVV-A can enhance the QoE under lightly loaded conditions. Even in the MVV-A system with MPEG-DASH, i.e., HTTP/TCP based transmission, the viewpoint change function affects users' satisfaction.

In future study, we need to refine the MVV-A system with MPEG-DASH and realize QoE-oriented MVV-A transmission with MPEG-DASH. In addition, we need to assess the effect of contents with further experiments with other contents. We also need to evaluate QoE for various network conditions and camera arrangements.

Acknowledgment

We thank Professor Emeritus Shuji Tasaka for his valuable discussion.

References

[1] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *Multimedia, IEEE*, vol. 18, no. 4, pp. 62-67, Apr. 2011.

[2] ISO/IEC 23009-1, "Dynamic adaptive streaming over HTTP (DASH) Part1: Media presentation description and segment formats," May 2014.

[3] I. Ahmad, "Multi-View Video: Get Ready for Next-Generation Television," *Proc. IEEE Distributed Systems Online*, vol. 8, no. 3, art. no. 0703-o3006, Mar. 2007.

[4] ITU-T Rec. P.10/G.100, Amendment 2, "New definitions for inclusion in Recommendation ITU-T P.10/G.100," July 2008.

[5] Y. Cao, X. You, J. Wang and L. Song, "A QoE friendly rate adaptation method for DASH," *Proc. IEEE BMSB 2014*, pp. 1-6, June 2014.

[6] Y. Liu, S. Dey, D. Gillies, F. Ulupinar and M. Luby, "User experience modeling for DASH video," *Proc. Packet Video Workshop (PV 2013)*, pp. 1-8, Dec. 2013

[7] H. Zhang, X. Gu and R. Ishibashi, "Seamless and efficient stream switching of multi-perspective videos," *Proc. Packet Video Workshop (PV 2012)*, pp. 31-36, May 2012

[8] E. Jimenez Rodriguez, T. Nunome and S. Tasaka, "QoE assessment of multi-view video and audio IP transmission," *IEICE Trans. Commun.*, vol. E92-B, no. 6, pp. 1373-1383, June 2010.

[9] "webm-dash-javascript," *https://chromium.googlesource.com/webm/webm-dash-javascript/*.

[10] Mindcraft Inc, "WebStone benchmark information," *http://www.mindcraft.com/webstone/*.

[11] "Apache HTTP SERVER PROJECT," *http://httpd.apache.org/*.

[12] "libwebm," *https://chromium.googlesource.com/webm/libwebm*.

[13] "webm-tools", *https://chromium.googlesource.com/webm/webm-tools*.

[14] "libvpx - The WebM Project," *http://www.webmproject.org/code/*.

[15] "Ogg Vorbis - Xiph.org," *http://xiph.org/vorbis/*.

[16] J. P. Guilford, *Psychometric methods*, McGraw-Hill, N. Y., 1954.

[17] "Wireshark," *http://www.wireshark.org/*.

[18] T. Nunome and T. Ishida, "Multidimensional QoE of multiview video and selectable audio IP transmission," The Scientific World Journal, Article ID 417290, 2015.