

DASH-based Streaming Client for View Switching in Free-viewpoint Video Systems

Seulki Song, Yong-Suk Park, Jungwook Wee

Korea Electronics Technology Institute

Seoul, Korea

Email : {seulkiemma, yspark, jwwee}@keti.re.kr

Abstract— Image processing and communications networking technologies are evolving at a rapid pace. Users are demanding high-quality and realistic services to enjoy cultural contents such as sports and concerts. Free-viewpoint video (FVV) streaming is an active field of research that enables realistic services. FVV streaming service augments the user experience by providing different viewpoints of the same scene. The user can navigate freely and continuously to any viewpoint. In order to provide FVV streaming services, the transmission method of various on-demand user viewpoints is important. In this paper, we propose and implement a DASH-based FVV streaming client to reduce the view switching latency by receiving a number of adjacent images. We verify the performance of the proposed client and analyze the switching time and transmission data rate depending on the number of simultaneously received videos.

Keywords— *Free-viewpoint video, FVV, DASH, HTTP, Streaming system*

I. INTRODUCTION

Multi-view video technology, which can provide various viewpoints in different directions using images acquired through a plurality of cameras, enables realistic multimedia services such as virtual reality (VR), super-multi view, and free-viewpoint video (FVV). FVV is a technology that allows users to view contents from a desired angle using images acquired from multiple cameras at various positions. This technology has recently attracted a great deal of attention in the field of sports, and interest in cultural performances is increasing. Although FVV provides the benefits of free-viewpoint user experience, its application to various services has been limited due to synthesis and network delivery of a large amount of image data.

However, due to the recent advances in video processing and transmission technologies, the enabling technologies for FVV, such as multi-view video codec, virtual view synthesis, and mass data transfer, are advancing rapidly. Especially, the method of synthesizing intermediate views on the client side by streaming adjacent images simultaneously has the advantage of reducing the latency of view switching requests, encoding, and

transmission compared to the method of synthesizing on the server side [1]. In order to transmit adjacent images, FVV streaming based on MPEG dynamic adaptive streaming over HTTP (DASH) has been proposed [2]. Free-viewpoint relationship description (FRD) architecture is used in the design of FVV [3]. The FRD provides a structure that organizes the relationship information between images in FVV [4]. By providing the FRD, clients can easily obtain the adjacent image information when the user moves the viewpoint.

In this paper, an FRD-based FVV streaming client is proposed and implemented to reduce the view switching latency by receiving a number of adjacent images. We verify the performance of the proposed client and analyze the view switching time and transmission data rate depending on the number of simultaneously received images.

II. DASH-BASED FVV SYSTEM

The FVV system based on DASH is constructed as shown in Fig. 1. First, videos of varying viewpoints are captured by cameras to provide the FVV service. For efficient transmission and rapid decoding, the captured videos are converted into segments through preprocessing, such as encoding and segmentation. The segments are stored on the server along with the MPD. The MPD defines the characteristics of the segment and the information related to the transmission. To provide FVV services, FRD structure is included in the MPD. The FRD defines the necessary information to enable the transfer of a suitable viewpoint video according to user requests. The client downloads the MPD with the FRD attached and acquires viewpoint information by parsing the MPD. With the obtained information, the client controls view switching by downloading the adjacent views.

Due to latency in content requests and transfers over HTTP, it is difficult to provide seamless view switching. Therefore, the proposed client simultaneously downloads and decodes adjacent views of the current view being played. As users move left or right, it is possible to switch smoothly, giving a panoramic view effect. The proposed client allows a smoother transition by adding real-time synthesizing modules and reduces the number of views downloaded.

This research was supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2017 [R2017030041, Experience Maximization Technology for Cultural Contents at Free Selection View].

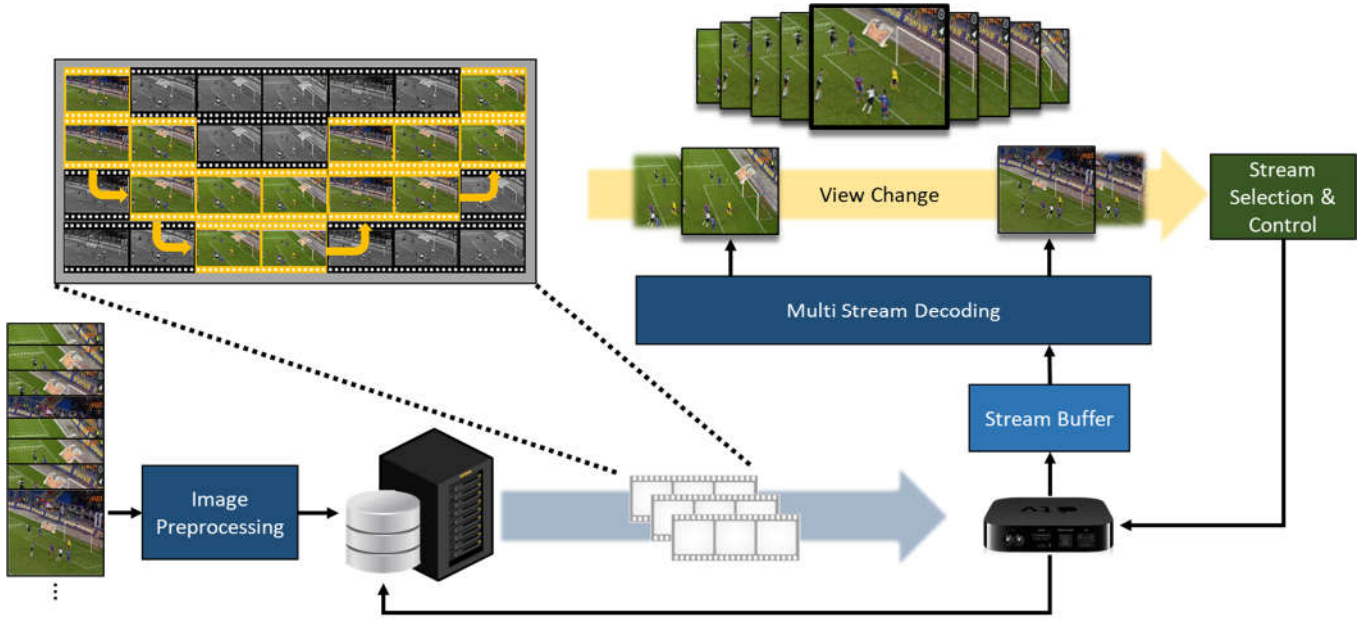


Fig. 1. Overview of the DASH-based FVV streaming system

III. DASH-BASED STREAMING CLIENT

The DASH-based streaming client requests MPD from the server. In order to enable navigation to different viewpoints, the client consists of HTTP Access Module, Buffer Manager, Media Processor, MPD Parser, Parameter Extraction, Stream Control Module, and View Selection, as shown in Fig 2. When metadata is extracted, the client obtains information such as the number of viewpoints in the FVV, the number of adjacent views, the absolute/relative position of the views, and the initial video. Using the acquired information, the client is set up in such a way, that it is free to move within the viewpoint provided by the FVV service. The initial view is played after setup. Depending on the network capacity, the client downloads multiple views simultaneously. The downloaded views are primarily synchronized through the buffer manager. The media processor decodes multiple views at the same time, but the renderer only plays the video of the view requested. To lower the latency in the user's viewpoint movement, the adjacent views are decoded simultaneously.

Fig. 3 shows an example of how the buffer manager is handled during view switching. The buffer manager is

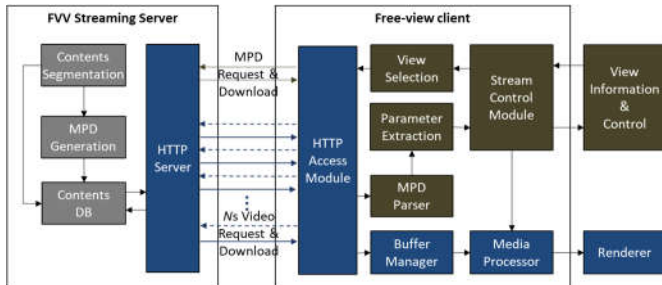


Fig. 2. Architecture of FVV server-client

buffering N views to the left and right of the main view requested by user. When the user moves the viewpoint, the path of the buffer is changed by the media processor. The newly added view is frame skipped to the present playback point in the media processor.

The implementation of the proposed client is shown in Fig. 4. The implemented client is able to change the point of view with the scroll button. There are buttons for providing multi viewpoint movement such as inward view, 360° view, and inward object view provided by FVV contents. Information about the view can be confirmed through the MPD parser viewer.

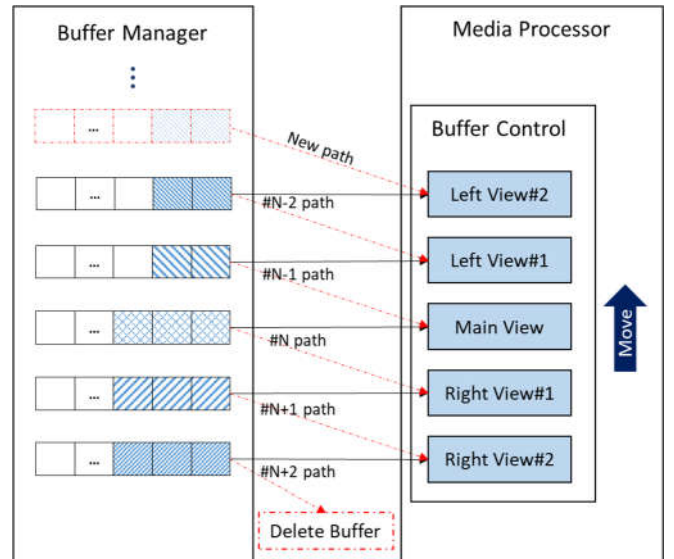


Fig. 3. Method of view switching in buffer manager



Fig. 4. DASH-based streaming client

IV. EXPERIMENTAL RESULT

A. Setup

We implemented the FVV system based on DASH. The client is implemented using C++ and a fully optimized proprietary HEVC decoder. To evaluate the proposed FVV client, we used the FVV content that we created and captured. The FVV content consists of three views: inward view, 360° view, and inward object view. The inward view is composed of 30 adjacent sub-views. We experimented with the inward view. All views are encoded in 1920x1080 resolution, 30 GoP, 30 fps, 3 Mbps bit rate, and 2-pass ABR method with HEVC. Coded videos were segmented using the open source GPAC framework, and MPD was created [5]. FRD structure with FVV content information was added to MPD. For performance comparison based on segment length, we set the length to 2 seconds and 4 seconds.

B. Results

We measured the time and the network rate when moving 50% of the total number of the views. We compared the performance to the average value because the speed differs depending on the position of the key-frame. For average measurement, we moved the fifteenth viewpoint every 5 seconds during playback. We experimented by increasing the number of simultaneous decoding to 5, 7, 9, 11, 13, and 15.

Fig. 5 shows the view switching time depending on the number of decoders. Even if the number of decoders increases, there is no difference in speed if the number exceeds 9. Fig. 6 shows the download rate of view switching depending on the number of decoders. As the number of decoders increases, the number of views downloaded and download rate increase. When compared by segment length, the difference in download rate depending on the length is not significant. However, the time of view switching becomes shorter when the segment length becomes as short as the key-frame interval.

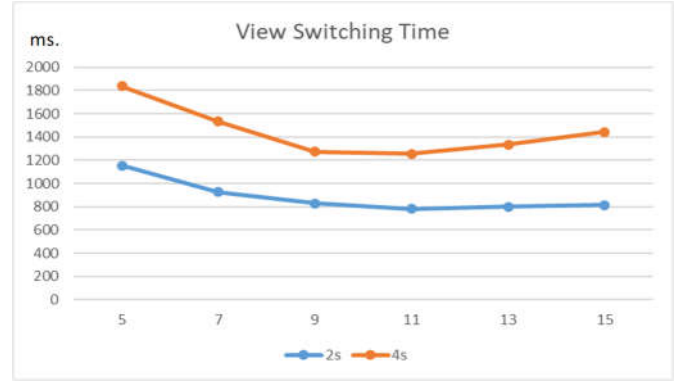


Fig. 5. View switching time with the number of decoders

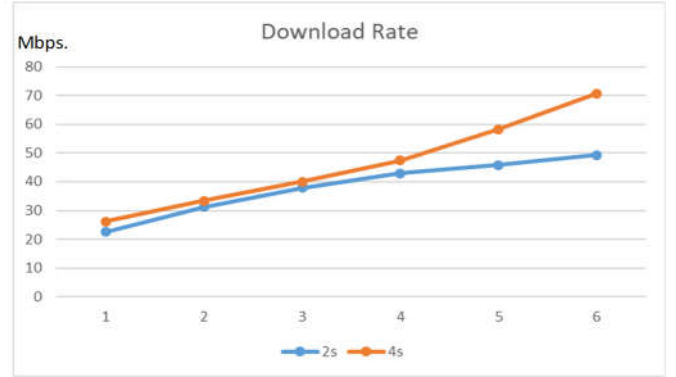


Fig. 6. Download rate of the number of decoders

V. CONCLUSIONS

In this paper, we proposed and implemented the DASH-based FVV streaming client to reduce the view switching latency by receiving a number of adjacent images. We verified the performance of the proposed client and analyzed the switching time and transmission data rate depending on the number of simultaneously received images. The proposed client shows optimal performance when 9 decoders are used simultaneously and the segment length becomes as short as the key-frame interval.

REFERENCES

- [1] A. Hamaza and M. Hefeeda, "Adaptive Streaming of Interactive Free Viewpoint Videos to Heterogeneous Clients," in Proc. 7th ACM Multimedia Syst. Conf, May 2016, Paper no. 10.
- [2] Information technology – Dynamic adaptive streaming over HTTP (DASH) – part 1 : Media presentation description and segment formats, 2016.
- [3] S. Song, Y. Kim, Y. Park, and J. Wee, "Free-viewpoint Relationship Description based Streaming Systems for Arbitrary View Switching" in Proc. ICUFN, July. 2018, pp.738-740.
- [4] S. Song, Y. Park and J. Wee, "Design of Free-viewpoint Relationship Description based on MPEG-DASH" in Proc. KSII ICONI, Dec. 2017, pp. 391-392.
- [5] GPAC multimedia framework. <http://gpac.wp.minews-telecom.fr/>.