Journal of Electrical Engineering 6 (2018) 33-39
doi: 10.17265/2328-2223/2018.01.005

Development of System for Simultaneously Present Multiple Videos That Enables Search by Absolute Time

Kazuhiro OTSUKI[1] and Yoshihiro FUJITA[2]

1. NHK Science & Technology Research Laboratories, Tokyo 157-8510, Japan

2. The Department of Electrical and Electronic Engineering and Computer Science, Ehime University, Ehime 790-8577, Japan

Abstract: We propose a system for simultaneously presenting numerous pieces of video content with absolute time metadata attached for effectively utilizing increasing number of pieces of video content. The proposed stored format used for the system is based on the MMT (MPEG (moving picture experts group) media transport) standardized method, which makes it possible to search by absolute time, and it is a format easy to access by HTTP. We implemented software that can simultaneously present multiple video files according to that format. The constructed system was able to provide service within a feasible processing time.

Key words: MMT, search by absolute time, system for simultaneously present multiple videos.

1. Introduction

Practical satellite-broadcasting of 8K SHV (Super Hi-Vision) will start in December of 2018, providing super-high-definition video and multi-channel audio included in 8K content. As the media transport scheme for 8K SHV, the MMT (MPEG (moving picture experts group) media transport) [1] was adopted [2]. MMT enables content to be distributed over multiple transmission paths. By using a UTC-based timestamp as the reference time in MMT, content can be transmitted synchronously. We expect to achieve full-scale broadcasting services that are enhanced with broadband networks, which include functions such as the ability to view multiple pieces of content at the same time, to select from multiple languages, and to view content across multiple devices (Fig. 1), as well as video services with a high sense of presence.

However, delivery services for user-generated video content, as represented by YouTube, are becoming more generalized in the field of broadband networks. In addition, SVOD (subscription video on demand) services have been introduced by content-delivery companies such as Netflix and Hulu, and video content sharing services by public users have appeared, such as Meerkat and Periscope. The annual global IP traffic will surpass the 2.3-zettabyte threshold, and 82 percent of all consumer Internet traffic will be video traffic, according to the Cisco Visual Networking Index forecast [3], by 2020. Video content that can be accessed from multiple locations and angles will tend to continuously increase.

In this paper, we propose a novel video presentation system for broadcast contents and user generated content. The rest of this paper is organized as follows. Details on MMT technology are given in Section 2. Then, the assumed total system and requirements are presented in Section 3. Formats stored in the system, the simple MPU format and software implementation are described in Sections 4 and 5. Evaluation results are shown in Section 6. Finally, Section 7 concludes the paper.

2. MMT

MMT is a standardized method that does not depend on transmission channels. We can display media with the same timing by decoding it on the basis

Corresponding author: Yoshihiro FUJITA, professor, research fields: media technology and applications.

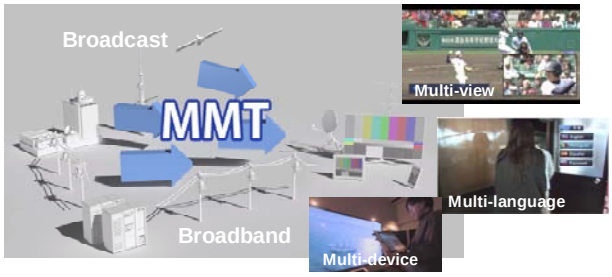

Fig. 1 Example of integrated broadcast and broadband services.

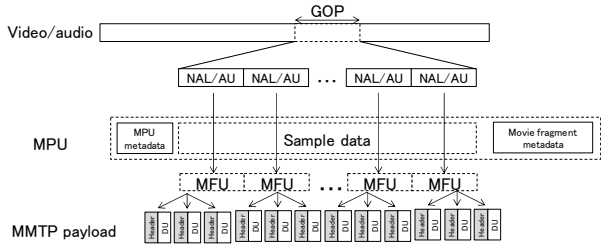

Fig. 2 Outline of configuration of MPUs.

of a coordinated universal time (UTC)-based timestamp described in the signaling information by using self-decodable units called “MPUs (media processing units)”. In other words, we can manage media to which tags of absolute time have been added.

MPUs include frame units of video, encoding units of audio, and data files used in applications. These units can decode video and audio by themselves. It is possible to extract smaller units called “MFUs (media fragment units)” by dividing MPUs. MFUs are units such as NAL (network abstraction layer) units, access units, or single files.

MFUs and MPUs are stored in an MMT protocol payload when transmission occurs. Metadata that constitute MPUs are not transmitted in the Association of Radio Industries and Businesses Standard (ARIB STD-B60) [4]. This is because the information required on the receiving side is described in the header of the MMTP (MMT protocol) payload, and it is therefore not necessary to transmit it again. Transmission is possible with this encapsulation with less overhead and simpler and lower delay. Fig. 2 outlines the configuration of an MPU that comes in a video/audio signal that is stored in the MMTP payload. This example has one MFU that is divided into three MMTP payloads. As IP (Internet protocol) packets are generally sized not to exceed 1.5 kilobytes, large MFUs are divided into multiple MMTP payloads.

The main feature of MMT is multi-protocol compliance. It is not only possible to support multicasting and unicasting with the UDP (user datagram protocol) but also the HTTP (hypertext transfer protocol) by using the TCP (transmission control protocol) [5]. The use of MMT as a distribution protocol in broadband networks has remained at the experimental trial level [6]. How to encourage service providers and others to use MMT is an issue that needs to be resolved to enable MMT to spread in the future.

3. System and Requirements

We studied the practical use of effective video content. Video content is extracted from spatial information at certain times and places in real time. We may discover other information that we have not been to by gathering information from multiple videos.

We assumed that there were innumerable amounts of video content in certain systems, regardless of whether the senders were broadcast companies, delivery companies, or general users. We would therefore be able to search scenes (of certain places) at certain times and to synchronize and play multiple scenes from the innumerable amounts of video content in these systems.

Various use cases are conceivable, such as those that offer related videos for news content, updated reports on accidents and disasters, multi-angle views, and replays of sports programs. The functional requirements to achieve such use cases are listed below.

(1) Inclusion of multiple broadcasting systems: A broadcasting system should function as part of a whole system.

(2) Easy upload of video contents: Video content should be uploaded to the network easily regardless of

the user or device.

(3) Search for a required scene: Video content should be handled in a unified form regardless of the sender, or in a form that can easily be interconverted. The video content should also have metadata on its time and location.

(4) Easy playback of video content: Search and playback of video content using metadata should be easy via a network.

Fig. 3 outlines the total system we assumed in an example of a sports program.

4. Stored Formats

We analyzed three existing formats used as stored formats, and compared them, and found that the format suitable for the proposed system was the simple MPU format [7]. A brief description of each format and a comparison table is shown.

4.1 PCAP

Fig. 4 outlines the PCAP (packet capture) format [8]. Each IP packet in the format stores the header of the timestamp that is given. This is possible by using this format to store a stream in real time so that it can easily be re-sent at the correct timing. When we search stored files for PTSs (presentation timestamps) to be used as a key, however, it is necessary to find signaling information in MMTP packets that are

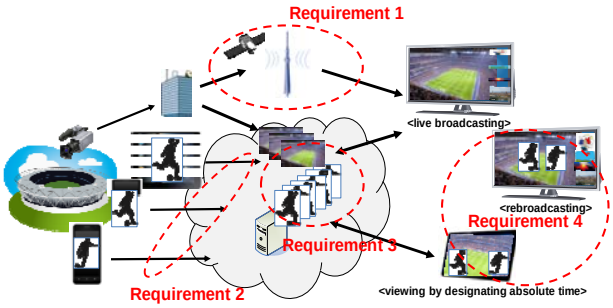

Fig. 3 Assumed total system.

Fig. 4 Packet capture (PCAP) format.

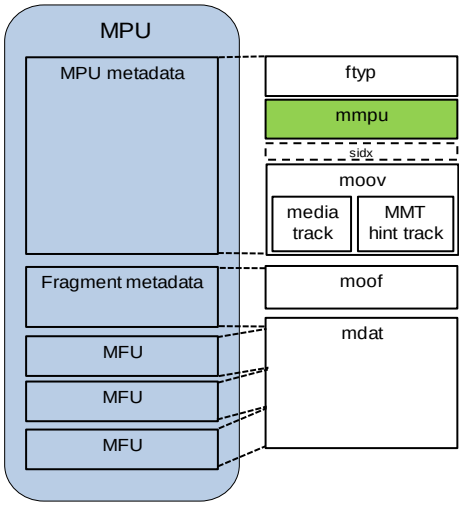

Fig. 5 ISOBMFF (ISO base media file format)-based MPU format.

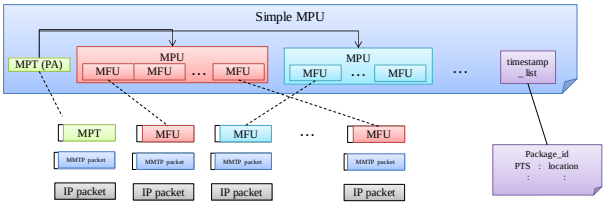

Fig. 6 Simple MPU format.

conveyed in IP packets, and to read the PTSs in the signaling information from an MPT (MMT package table).

4.2 ISOBMFF (ISO Base Media File Format)

Fig. 5 outlines the ISOBMFF-based MPU format. A file in this format has a box structure according to an established standard. It is possible to search for stored files for PTSs to be used as a key by referring to the content of a specific box and accessing the required MPU by using the box unit. However, we cannot directly store a real-time stream for ISOBMFF. It is necessary to convert the box structure when storing information. In addition, it is necessary to pass a parameter where the box structure has to be converted to a stream format when we re-send a stored file.

4.3 Simple MPU

MPUs in the proposed format (Fig. 6), i.e., simple MPU format, are extracted from the packets of real-time transmission by UDP and stored in a

location by using a source address and a packet ID. Control-information (MPT)-changed location descriptions are stored at the same time. In addition, meta-information (timestamp_list), i.e., a combination of UTC-based PTSs and the location of each MPU, is stored. As a result, it becomes possible to immediately acquire the required MPU by using HTTP and searching with PTS as a key.

4.4 Comparison

Table 1 compares the three formats. The PCAP format is suitable for storing ARIB profiles and UDP transmission, but it is not suitable for scene search or HTTP access performed by designating absolute times. However, the ISOBMFF-based MPU format is not suitable for storing ARIB profiles or UDP transmission, but it is suitable for scene search and HTTP access performed by designating absolute times. Storing ARIB profiles and UDP transmission are possible with the simple MPU format, and it is suitable for scene search and HTTP access performed by designating absolute times.

As can be seen from the comparison table, the PCAP method is mainly suitable for broadcasting services centered on live delivery with low delay. The simple MPU method, in comparison, is suitable for services that perform scene search at absolute times and HTTP access, as we assume in this paper.

Table 1 Comparison of stored formats.

	PCAP	ISOBMFF-based MPU	Simple MPU format
Storage of ARIB profiles	✓[a] (storage of IP packets)	×[b] (need to add metadata)	Δ[c] (need to generate original metadata)
UDP transmission	✓ (possible to transmit data as they are)	× (inverse conversion required)	Δ (possible by MMTP conversion)
Scene search by absolute time	× (analyze IP packets)	Δ (analyze MPU metadata)	✓ (analyze timestamp_list)
HTTP access	× (difficult)	Δ (possible)	✓ (easy)

a. Suitable
b. Not suitable
c. Conditionally suitable

5. Software Implementation

We implemented conversion software and a scene search script with the simple MPU method. The conversion software accumulates MMT real-time streams and video files from smart devices with this method. Video and audio MPUs from an MMT stream and MPT files used as signaling information to describe the general location of video and audio are stored. The packet ID of all media is described as a general location when broadcasting, but it is rewritten in the conversion software to the location of MPUs placed on the HTTP server and stored. Furthermore, timestamp_list that lists pairs of UTC-based PTSs and the location of each MPU are generated. Similarly, for video files from smart devices, video and audio MPUs and timestamp_list are stored. In this case, the MPU file is in the unit of a GOP (group of picture) and ADTS (audio data transport stream). The PTS described in the timestamp_list is a time added to the recording start time, an offset value obtained from the duration (recording time), the number of GOPs of stored video, and the number of ADTS frames of audio. Fig. 7 outlines functional configuration diagrams of the server sides.

The scene search script runs on an HTTP server that enables HTTP access by designating absolute times to the files stored by using the simple MPU format.

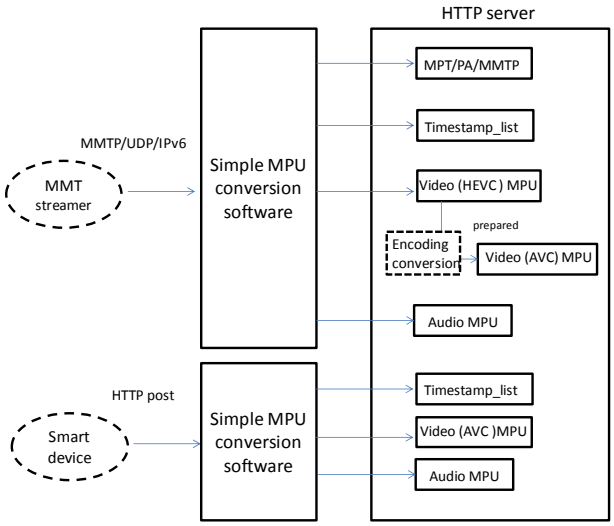

Fig. 7 Functional configuration for server side.

Timestamp_lists on the server are searched according to the absolute time, the number of search scenes and search duration that is requested by the client. This script sets the absolute time as the start time, selects only media that contains search scenes with PTSs close to the start time existing within the search duration period, and responds to the client in the form of MPD_list.

We currently group candidates of timestamp_list using the source address, which indicates differences between cameras, and the package ID, which indicates differences between programs with an algorithm for selecting candidates close to the required absolute time. Nevertheless, it will be necessary to consider more advanced methods in the future, such as those using GPS data that indicate the location of the scene.

Client software needs to be able to present multiple scenes from multiple simple MPU files placed on the HTTP server by designating absolute times requested by the client. It seems preferable from the viewpoint of a user interface to be able to simultaneously view multiple screens. In addition, since only one HEVC (high efficiency video coding) video is decoded due to current terminal capabilities, the client software switches between multiple video images to be reproduced at the same time reference for display screens that correspond to the reception of multiple video images.

Therefore, we implemented clients that make it possible to simultaneously view multiple screens with a browser by converting HEVC to AVC (advanced video coding) and the simple MPU method to MPEG-Dynamic adaptive streaming over HTTP (DASH) [9]. Fig. 8 outlines a functional configuration obtained with the MPEG-DASH method.

A browser is used in the MPEG-DASH method as the client software. The MPU of the video is converted beforehand on an HTTP server from HEVC to AVC as it can be played in the browser. In response to a request by the client, MPD_list, the MPD (media presentation description) and DASH segments are

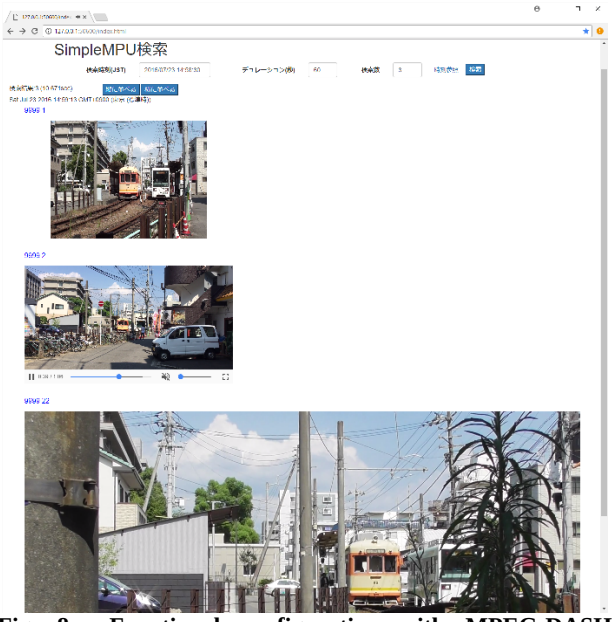

Fig. 8 Functional configuration with MPEG-DASH method.

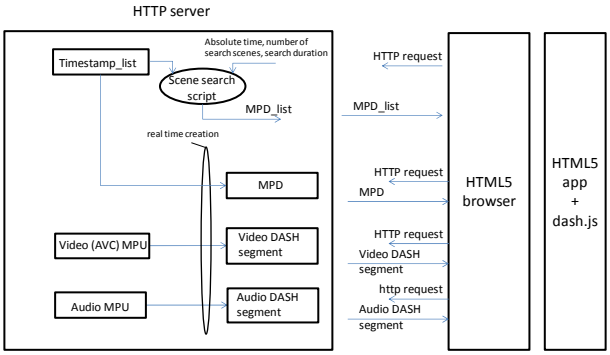

Fig. 9 Display screen on HTML5 browser.

generated in real time and are returned to the client. The browser is a conventional HTML5 browser, and simultaneous views of multiple scenes are achieved with the HTML5 application and dash.js acquires MPD and DASH segments according to the MPD_list. Fig. 9 shows photographs of the display screen on the HTML5 browser. We confirmed that multiple screens can be displayed at the same time, and that more intelligible video presentation becomes possible.

6. Evaluation

We evaluated the feasibility of using the implemented system. We considered the time before enjoying the service as one measure of the feasibility

of services. That is, if the time to upload content or the time to search and present the content takes a long time, the service will not be used by customers. Immediacy is a very big factor in considering the feasibility of services.

First, since the stream data sent from the MMT streamer are converted in real time, it takes time to process only the real time of the content. However, with regard to user generated content recorded by a smart device, it is necessary to convert the stored file. Therefore, the processing time according to the length of the content was measured. Table 2 shows the results.

As shown in the results, it is possible to change processing in about half the length of the content and it can be said that conversion to the simple MPU format is within a feasible range.

Next, the time from a request from the client to the presentation of the result was also measured. Table 3 shows the results.

It seems that there was a difference in drawing processing depending on the capability of the client device. In general, as the reproduction time increases and as the number of presented videos increases, the processing time of presentation is required.

Regarding the reproduction time, the proposed system is thought to be suitable for service applications such as searching for videos one after another within a short playback time rather than continuing to watch for a long time.

Regarding the number of presented videos, it is considered difficult to watch multiple videos at the same time as the number increases, and it is practical to display about four videos. From the results, videos can be provided within a feasible processing time.

From the above, the proposed system with the simple MPU format can be sufficiently realized when used for services that it is suited for.

Table 2 Processing time of upload.

Length of content	10 s	20 s	30 s	60 s
Processing time of upload	4.77 s	8.62 s	12.82 s	25.31 s

Table 3 Processing time of presentation.

Reproduction time	15 s	30 s	45 s	60 s
2-video presentation	2.57 s	4.71 s	7.19 s	8.73 s
3-video presentation	3.87 s	7.92 s	13.18 s	20.03 s
4-video presentation	5.22 s	9.90 s	14.88 s	25.96 s

7. Conclusion

We proposed a system that effectively utilizes innumerable pieces of video content. We compared three formats for required stored formats, including one based on MMT, which is a media transport method for SHV satellite broadcasting. We demonstrated that the format was suitable for the assumed system and tested and verified its performance through software implementation. We intend to upgrade the search algorithms of the system in the future.

References

[1] ISO/IEC. 23008-1:2014: Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 1: MPEG Media Transport (MMT).

[2] Otsuki, K., Aoki, S., Kawamura, Y., Tsuchida, K., and Kimura, T. 2014. “MMT-based Super Hi-Vision Satellite Broadcasting Systems.” *NAB Broadcast Engineering Conference Proceedings*, 38-44.

[3] Cisco Visual Networking Index: Forecast and Methodology. 2015-2020 White Paper. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html.

[4] ARIB STD-B60 (Version 1.8): MMT-based Media Transport Scheme in Digital Broadcasting System. September 2016 (in Japanese).

[5] Aoki, S., Kawamura, Y., Otsuki K., and Hashimoto, A. 2016. “A Study on Conversion from MMT Protocol to HTTP.” The Institute of Electronics, Information and Communication Engineers, Communications Society Conference B-6-10 (2016) (in Japanese).

[6] Kawamura, Y., Otsuki, K., Hashimoto, A., and Endo, Y. 2016. “Functional Evaluation of Hybrid Content Delivery Using MPEG Media Transport.” *IEEE International Conference on Consumer Electronics 2016*, 267-8.

[7] Otsuki, K., Hayami T., and Fujita,Y. 2016. “A Study of Stored Formats to Enable HTTP Access by designating

Absolute Time." *ITE Technical Report* 40 (45): 17-20.

[8] Pcap Definitions [WinPcap user's manual]. https://www.winpcap.org/docs/docs_412/html/group__wpcap__def.html.

[9] ISO/IEC 23009-1:2014: Information Technology—Dynamic Adaptive Streaming over HTTP (DASH) —Part 1: Media Presentation Description and Segment Formats.

Kazuhiro Otsuki received his M.E. degrees in science and engineering from the Tokyo Institute of Technology in 1997. He joined NHK in 1997 and has since been researching and developing digital broadcasting system at the Science and Technology Research Laboratories of NHK.

Yoshihiro Fujita is currently a Professor at Ehime University. From 1976 to 2010 he was with NHK (Japan Broadcasting Corporation). He conducted research on advanced imaging systems, HDTV and UHDTV broadcasting systems at NHK Science and Technology Research Laboratories (NHK STRL). From 2006 to 2010, he was the Director of System Research Division where he was in charge of broadcasting communication integration systems, and then the Deputy Director of the Laboratories. He received his B. S. and PH.D. Degrees in 1976 and 1998 respectively from the University of Tokyo and is a fellow of IEEE and ITEJ (Institute of Television Engineers Japan).