Free-viewpoint Video Communication Using Multi-view Video Coding

Hideaki Kimata†*, Masaki Kitahara, Kazuto Kamikura, and Yoshiyuki Yashima*

Abstract

We propose a video communication method that lets the user freely change his/her viewing position and viewing direction. Free-viewpoint video is a next-generation visual application that has been extensively studied and discussed by the international MPEG standard body as 3DAV (three-dimensional audio and visual). For this application, we propose a multi-view video coding method and communication protocol. We also describe a free-viewpoint video viewer that generates a natural view from an arbitrary viewing position and viewing direction, using decoded multi-view video data and generated image data by the Ray-Space interpolation and extrapolation methods.

1. Introduction

Free-viewpoint video is expected to be a next-generation visual application [1]. It provides the user with realistic impressions by means of high interactivity and photorealistic image quality. For instance, it lets the user freely change his/her viewpoint (i.e., viewing position and viewing direction) and enjoy more photorealistic three-dimensional (3D) images. With these functionalities, it will be used for various services, such as broadcasting, visual communication, and education.

Two main technical coding areas are involved in free-viewpoint video [2]: multi-view video coding, and view generation [3]. Generally, video signals captured by two or more cameras are used as source data. Multi-view video coding is used to code these video signals. Views that were not captured can be generated in the viewer. This view generation technique can be used to generate image signals for displays and for multi-view video coding [4].

Using the concept of plenoptic*1 functions addressed by Adelson and Bergen [5], free-viewpoint video can be represented by light rays in seven-dimensional space. Chai et al. proposed a plenoptic-sampling analysis and a minimum sampling density that does not cause aliasing artifacts based on two assumptions: the artifacts from the occlusion can be neglected, and the bidirectional scatter distribution function (BRDF) model is Lambertian [6]. View generation is a kind of upsampling technique for seven-dimensional space defined by the plenoptic function. It can be used for sampling densities smaller than Chai's minimum sampling density, in addition to larger ones. Various kinds of view generation techniques have been proposed, such as interpolation/extrapolation methods [7], [8], depth-based methods [9], and model-based methods such as multi-texture [10]. The interpolation/extrapolation method is the most straightforward one because it applies sample features in the space defined by the plenoptic function or its subset. Ray-Space is a practically definable space, rather than a conceptual space defined by the plenoptic function. It was originally proposed by Fujii et al. [11]. Interpolation/extrapolation filters for Ray-Space, which cover the occlusion areas, have

† NTT Cyber Space Laboratories
Yokosuka-shi, 239-0847 Japan
E-mail: kimata.hideaki@lab.ntt.co.jp

*1 plenoptic function: a single function that describes the structure of the information in the light impinging on an observer. Since this function describes everything that can be seen, Adelson and Bergen call it the plenoptic function (from *plenus*, complete or full, and *optic*).

been proposed, and discussed [11]. The surface light field method, a coding technique for texture information of 3D meshes, is an extension of Ray-Space [12].

In this paper, we propose a free-viewpoint video communication method, which includes a video coding method and communication protocol. This method lets the user freely change his/her viewpoint, while downloading or streaming visual data. This method is not a form of broadcasting, i.e., it does not transmit all the visual data. Therefore, the QoS (quality of service) of the visual data can be as high as possible with the available communication bandwidths. While previous research concentrated on view generation techniques more than coding methods, we studied coding methods more because our target application is communication. Technically this method uses multi-view video coding and Ray-Space-based view generation techniques.

Section 2 gives an overview of free-viewpoint video communication, section 3 introduces a prototype of the free-viewpoint video viewer, section 4 describes the coding method, and section 5 describes communication protocols.

2. Free-viewpoint video communication

In free-viewpoint video communications, a user can watch a video while freely changing the viewing position as illustrated in **Fig. 1**. This application can be used for both on-demand and live broadcasts. Even in a live broadcast, we need a huge memory for storing video data, unlike a conventional two-dimensional (2D) video broadcast. This is because this application does not transmit all video data to the users even in a live broadcast. We assume that the video content of the application is composed of multi-view video data and possibly some information for view generation. Therefore, some of the multi-view video data can be transmitted through a network to the user. Multi-view video data is a highly calibrated group of views. The camera parameter estimations (camera calibrations) are completed before views for the video content are captured. Rectification of the captured camera images is completed before compression.

This application requires two main functions: QoS-guaranteed transmission of video data in the available bandwidth and low-delay random access in terms of time stamp and viewing position. Time-stamped random access is the same requirement as in conventional 2D video streaming applications, but random access in terms of viewing position is a specific requirement for this application.

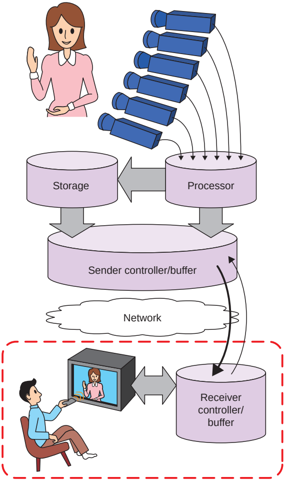


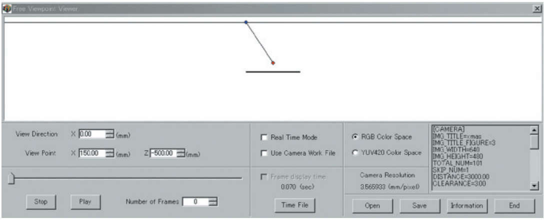
Fig. 1. Free-viewpoint video communication.

3. Free-viewpoint video viewer

We developed a prototype free-viewpoint video viewer. This viewer generates an image from multiple sets of video data stored in the storage device and displays it when the user changes his/her viewing position. It does not have a video decoding engine. Its role in the communication system is indicated by the red dotted square in Fig. 1.

This viewer was developed as software on a PC. It is composed of a controller (**Fig. 2**) and a display. The user can interact with the video content, changing his/her viewpoint with a click of the mouse at the desired viewpoint.

The input multiple video data is composed of a series of uncompressed pictures, which were captured earlier by multiple cameras arranged in a horizontal line (**Fig. 3**). The intervals between adjacent



Free Viewpoint Viewer

View Direction X 0.00 (mm)

View Point X 150.00 (mm) Z -500.00 (mm)

□ Real Time Mode

□ Use Camera Work File

● RGB Color Space

○ YUV420 Color Space

```
[CAMERA]
IMG_TITLE=xmas
IMG_TITLE_FIGURE=3
IMG_WIDTH=640
IMG_HEIGHT=480
TOTAL_NUM=101
SKIP_NUM=1
DISTANCE=3000.00
CLEARANCE=3.00
```

□ Frame display time 0.070 (sec)

Camera Resolution 3.565933 (mm/pixel)

Number of Frames 0

Fig. 2. Free-viewpoint video viewer controller.

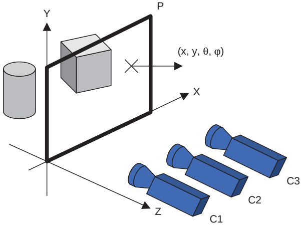


Fig. 3. Camera arrangement supported by the viewer.

cameras are all the same. The supported color spaces are RGB (red, green, blue) and YUV 4:2:0.

Here, we explain the view generation method we used. The algorithm is based on the Ray-Space approach [11]. All the originally captured views are arranged regularly in the Ray-Space. We define light rays across the 2D plane illustrated in Fig. 3. The sample value of a light ray is a four-dimensional value (x, y, θ, φ), where (x, y) is the cross position and (θ, φ) is the cross angle of the light ray. **Figure 4(b)** illustrates the (x, u) cross section image, which corresponds to the camera arrangement (positions C1, C2, and C3), where u is defined as u=tan θ. The arrows in Fig. 4(a) indicate the viewing directions. The generated-view lines V1, V2, and V3 in Fig. 4(b) indicate the positions of the samples in the Ray-Space. They are equivalent to the real positions V1, V2, and V3, respectively in Fig. 4(a). Viewing directions of V1 and V2 are the same as the camera directions. However, viewing direction of V3 is not the same as any camera direction. For view generation, the values of all the samples in a generated-view line need to be calculated using values of already available samples in the original-view lines. In particular, as shown for V2 and V3, the samples for calculation are involved in more than two original-view lines.

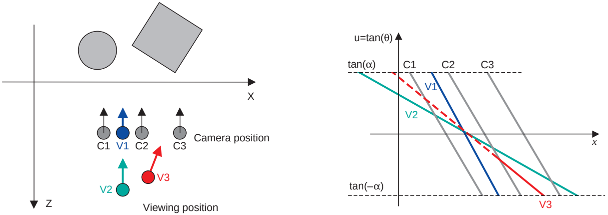
To speed up the loading of necessary image information for view generation, the arrangement of the samples in the picture is different from the state during capture. To generate views, the views that contain used samples are the same for the vertical direction. However, they are different for the horizontal direction, depending on the generated sample position. Therefore samples in a picture are rearranged to the positions where their relative positions are rotated by 90°, so that the viewer can load the necessary samples sequentially, as illustrated in **Fig. 5**. This rearrangement might be useful for speeding up the decoding of multi-view video data, if the views were encoded by the conventional 2D video coding method, because the samples are basically encoded in a raster scan order.

The viewer displays the generated view without needing any special graphics hardware.

4. Coding method

Here, we describe our multi-view video coding method [13], [14]. We designed this method by taking into account the two requirements discussed in section 2. This method is similar to the disparity and motion compensated compression algorithm for simplified dynamic light fields (SDLFs) [15]. However, our method is much more suitable because it provides random access.

For low-delay random access, we introduce the



(a) Camera positions

(b) Cross section image

Fig. 4. Camera positions and cross section image.

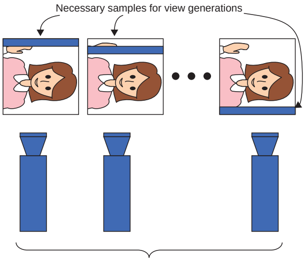
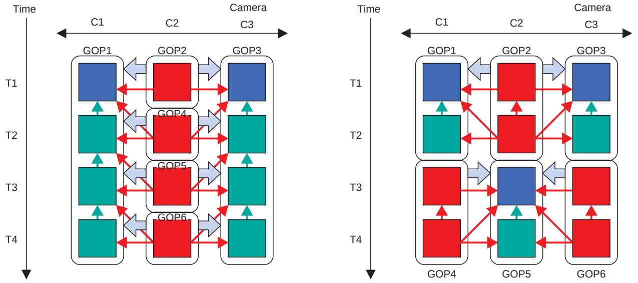


Fig. 5. Sample arrangement in a picture.

concept of a group of GOP (GoGOP), which is an extension of a group of pictures (GOP). Conventional 2D video coding schemes normally define GOP to provide random access in terms of the time stamp. GoGOP provides low-delay random access, in terms of viewing position as well as time stamp. In GoGOP, GOP is defined within a view, and all GOPs are categorized into two kinds, base GOP and inter GOP. A picture in a base GOP may use decoded pictures only in the current GOP. However, a picture in an inter GOP may use decoded pictures in other GOPs as well as in the current GOP. The current GOP index and the reference GOP index are defined in the GOP header. The reference GOP index indicates the GOP whose decoded pictures may be used for decoding pictures in the current GOP. Two typical examples of the GoGOP structure are illustrated in **Fig. 6**, where a blue square indicates an intra coded picture, a green square indicates an inter coded picture that refers only to the same GOP, and a red square indicates an inter coded picture that refers to another available GOP. Either base GOP or inter GOP can be set within a view, as illustrated in Fig. 6(a). In this case, GOP2 refers to GOP1 and GOP3, and GOP4 and GOP6 refer to GOP5. Even if inter GOPs are not decoded, all views can be obtained by decoding base GOPs, although obtained GOPs are temporarily subsampled. Figure 6(b) illustrates a structure that achieves low-delay access to a view, because every picture in an inter GOP uses only decoded pictures in base GOPs. This structure guarantees no delays while base GOPs are decoded on time. In this case, an inter GOP contains only one picture.

To decode a GoGOP bitstream, we introduce the hierarchical reference picture selection (HRPS) method [16]. **Figure 7** illustrates the decoder configuration. This method is an extension of MPEG-4 advanced video coding (AVC). In particular, the decoder has a layered structured reference picture memory. Decoded pictures in one GOP are stored in one layer in that memory. The reference picture indices adaptively correspond to the reference pictures in multiple layers.

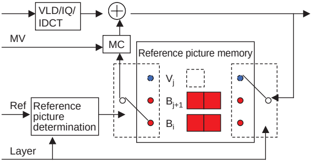
Moreover, to control the bitrate of the video content, we introduce a wavelet-based sub-band structure to a picture. All pictures are transformed with wavelet filters, such as JPEG2000. Only the LL band coefficients are coded using HRPS. The other band data is adaptively truncated to control the bitrate. This coding scheme is illustrated in **Fig. 8**.



(a) Structure of GoGOP #1

(b) Structure of GoGOP #2

Fig. 6. Structure of GoGOP.

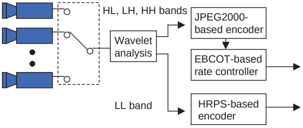


MV: motion vector
MC: motion compensation
VLD: variable length decoding
IQ: inverse quantization
IDCT: inverse discrete cosine transformation

Fig. 7. Decoder configuration of HRPS.

5. Communication protocol

In free-viewpoint video communication, a receiver indicates his/her viewpoint. These feedback messages are transmitted to the sender over a backward channel. If the network is an IP-based packet network, then RTCP is an appropriate protocol because it can control multiple video data streams transmitted using RTP (RTP: real-time transport protocol, RTCP: RTP control protocol). We transmit multiple video data as a single elementary stream. A sender multiplexes data for several views having the same time stamp into one RTP packet and transmits it. At the beginning of a session, RTSP can be used to indicate the initial viewpoint.



EBCOT: embedded block coding with optimized truncation

Fig. 8. Sub-band-based coding scheme.

Since there is a round trip delay between the receiver and sender, a viewing position prediction scheme, which calculates the viewpoint after the round trip delay is necessary in the receiver. Based on the results of this viewing position prediction, the receiver indicates to the sender the viewpoint associated with the time stamp (**Fig. 9**). This process is a kind of prefetch [17], because the receiver requests multi-view video data based on the required viewpoint and time stamp.

6. Conclusion

We proposed a novel free-viewpoint view communication scheme. This application allows the user to change his/her viewpoint freely while receiving video content. It requires two functions: QoS-guaranteed transmission of video data in the available band-

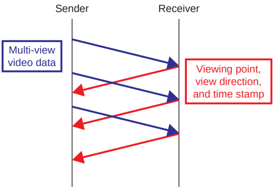


Fig. 9. Communication protocol.

widths and low-delay random access in terms of time stamp and viewing position. To achieve these functions, we propose a novel multi-view video coding method and communication protocol. We also described a developed prototype of the free-viewpoint video viewer. This viewer can generate a view from an arbitrary viewpoint, using Ray-Space interpolation and extrapolation methods.

References

[1] "Applications and Requirements for 3DAV," document N5877 MPEG Meeting, Trondheim, Norway, July 2003.

[2] "Report on 3DAV Exploration," document N5878 MPEG Meeting, Trondheim, Norway, July 2003.

[3] H. Kimata, "Preliminary study on multiple view coding for the Ray Space representation (3DAV EE2)," document M10054 MPEG Meeting, Brisbane, Queensland, Australia, Oct. 2003.

[4] M. Tanimoto and T. Fujii, "Ray-Space Coding Using Temporal and Spatial Prediction," document M10178 MPEG Meeting, Brisbane, Queensland, Australia, Oct. 2003.

[5] E. H. Adelson and J. Bergen, "The plenoptic function and the elements of early vision," in Computational Models of Visual Processing, Cambridge, MA: MIT Press, pp. 3-20, 1991.

[6] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, "Plenoptic sampling," in Proc. ACM Annu. Computer Graphics Conf., pp. 307-318, July 2000.

[7] T. Kobayashi, T. Fujii, T. Kimoto, and M. Tanimoto, "Interpolation of Ray-Space Data by Adaptive Filtering," IS&T/SPIE Electronic Imaging 2000, 2000.

[8] S. J. Gortler, R. Grzesczuk, R. Szeliski, and M. F. Cohen, "The Lumigraph," SIGGRAPH'96, pp. 43-54, 1996.

[9] J.-R. Ohm and K. Müller, "Incomplete 3D Representation of Video Objects for Multiview Applications," PCS'97, 1997.

[10] P. Debevec, Y. Yu, and G. Borshukov, "Efficient View-dependent Image-based Rendering with Projective Texture Mapping," Proc. ACM SIGGRAPH'98, Orlando, FL, U.S.A., July 1998.

[11] T. Fujii, T. Kimoto, and M. Tanimoto, "Ray Space Coding for 3D Visual Communication," PCS'96, pp. 447-451, 1996.

[12] M. Levoy and P. Hanrahan, "Light Field Rendering," Proc. ACM SIGGRAPH'96, pp. 31-42, Aug. 1996.

[13] H. Kimata, "Preliminary results on multiple view coding for the sparse Ray Space representation (3DAV EE2.2.1)," document M10327 MPEG Meeting, Waikoloa, Hawaii, U.S.A., Dec. 2003.

[14] H. Kimata, "Preliminary results on inter-view coding for the dense Ray Space representation (3DAV EE2.2.1)," document M10652 MPEG Meeting, Munich, Germany, Mar. 2004.

[15] H.-Y. Shun, S. B. Kang, S.-C. Chan, "Survey of Image-Based Representations and Compression Techniques," IEEE Trans., Circuits Syst., Video Technol., Vol. 13, No. 11, pp. 1020-1037, 2003.

[16] H. Kimata, M. Kitahara, K. Kamikura, and Y. Yashima, "Hierarchical Reference Picture Selection Method for Temporal Scalability beyond H.264," ICME 2004, Taipei, Taiwan, 2004.

[17] M. Reisslein and K. Boss, "A Join-the-Shortest-Queue Prefetching Protocol for VBR Video on Demand," In Proc. IEEE International Conference on Network Protocols, 1997.

Hideaki Kimata

Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.

He received the B.E. and M.E. degrees in applied physics from Nagoya University, Nagoya, Aichi in 1993 and 1995, respectively. In 1995, he joined NTT Human Interface Laboratories, where he has been engaged in R&D of low bitrate video coding and error resilient video coding algorithms, and visual communication systems. His research interests also include free viewpoint video coding and pre- and post-processing for video coding. He acts as a co-chair of MPEG 3DAV AHG. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan (IEICE), and the Institute of Image Information and Television Engineers of Japan (ITE).

Masaki Kitahara

Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.

He received the B.E. and M.E. degrees in industrial and management systems engineering from Waseda University, Tokyo in 1999 and 2001, respectively . He joined NTT in 2001 and has been engaged in R&D of model-based data compression for image based rendering and H.264 video compression algorithms. His research interests include signal processing methods for 3D applications, data compression, and hard-ware-assisted rendering algorithms for image-based rendering, and mesh parameterization. He is a menber of IEICE.

Kazuto Kamikura

Senior Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.

He received the B.E. and M.E. degrees in electrical engineering from Tokyo Science University, Tokyo in 1984 and 1986, respectively. He received the Ph.D. degree in electrical engineering from the University of Tokyo, Tokyo in 2000. He joined NTT in 1986 and has been engaged in R&D of video coding systems. His current research interests include digital image processing and video sequence coding. He is a member of IEICE and ITE.

Yoshiyuki Yashima

Senior Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.

He received the B.E., M.E., and Ph.D. degrees in electronics engineering from Nagoya University, Nagoya, Aichi in 1981, 1983, and 1998, respectively. In 1983 he joined the Electrical Communications Laboratories, Nippon Telegraph and Telephone Public Corporation (now NTT), Kanagawa, where he has been engaged in R&D of high-quality HDTV signal compression, MPEG video coding algorithms, and lossless image coding systems. His research interests also include pre- and post-processing for video coding, processing of compressed video, compressed video quality metrics, and image analysis for video communication systems. He is a member of the IEEE Signal Processing Society, the Information Processing Society of Japan, IEICE, and ITE.