Real-Time Video-Based Rendering for Augmented Spatial Communication

Takeshi Naemura and Hiroshi Harashima

Dept. of Inform. & Commun. Eng., The Univ. of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, JAPAN

ABSTRACT

In the field of 3-D image communication and virtual reality, it is very important to establish a method of displaying arbitrary views of a 3-D scene. It is sure that the 3-D geometric models of scene objects are very useful for this purpose, since computer graphics techniques can synthesize arbitrary views of the models. It is, however, not so easy to obtain the models of objects in the physical world. In order to avoid this problem, a new technique, called image-based rendering, has been proposed for interpolating between views by warping input images, using depth information or correspondences between multiple images. To date, most of the works on this new technique has been concentrated on static scenes or objects. In order to cope with 3-D scenes in motion, we must establish the ways of processing multiple video sequences in real-time, and constructing accurate camera array system. In this paper, the authors propose a real-time method of rendering arbitrary views of 3-D scenes in motion. The proposed method realizes a sixteen camera array system with software adjusting support and a video-based rendering system. According to the observer's viewpoint, appropriate views of 3-D scenes are synthesized in real-time. Experimental results show the potential applicability of the proposed method to the augmented spatial communication systems.

Keywords: 3-D Image Communication, Virtual Reality, Multiview Image, Camera Array, Image-Based Rendering, Video Texture, Light Ray Data, 3-D Space in Motion

1. INTRODUCTION

In the field of 3-D image communication and virtual reality, 3-D display technologies have been making rapid progress. It is well known that the binocular parallax, motion parallax and focal accommodation effects are essential for providing the 3-D visual effects. In order to produce the binocular and motion parallax, we need two views of a 3-D scene corresponding to left and right eyes. We can produce even the focal accommodation effect by providing several views for each eye.[1] Consequently, it is very important to establish a method of synthesizing arbitrary views of a 3-D space.

It is sure that the 3-D geometric models of scene objects are very useful for this purpose, since computer graphics techniques can synthesize arbitrary views of the models. In the field of virtual reality, the virtual world consists of geometric models of objects. In order to apply the same strategy for the physical world, we must construct geometric models of real objects. It is, however, not so easy to measure the accurate 3-D shapes, surface textures and reflection models of real objects.

In order to avoid this problem, a new technique, called image-based rendering, has been proposed. Most of the works on this new technique can be classified as follows :

Environment Map[2] and QuickTimeVR[3] An environment map records the incident light arriving from all directions at a point. This technique can be used to quickly display any outward looking view of the environment from a fixed location but at a variable orientation.

View Interpolation[4–6] Several techniques are proposed for interpolating between views by warping input images, using depth information or correspondences between multiple images.

Light Ray Data Space[7–14] Since image data is a set of light ray data, image capturing can be regarded as a sampling process of a data space, in which the light ray data is stored. Any image processing, including interpolation and rendering, can be done in the data space domain. This approach allows much more freedom in the range of possible views.

Further author information : naemura@hc.t.u-tokyo.ac.jp, http://www.hc.t.u-tokyo.ac.jp/~naemura

To date, most of the works on this new technique has been concentrated on static scenes or objects. This is because it is not so easy to capture and process multiple images in real-time. For the purpose of capturing multiple images of a scene at once, a camera array system can be utilized. However, there still remains several problems caused by the characteristics of cameras. For example, the directions of optical axes of cameras are slightly different from each other. Even if we can obtain multi-view video sequences of a 3-D scene, the process of view interpolation is too heavy to render virtual views in real-time.[6]

In order to cope with a 3-D space in motion, we must establish the ways of processing multiple video sequences in real-time, and constructing accurate camera array system. In this paper, the authors propose a real-time method of rendering arbitrary views of 3-D scenes in motion. This technology will contribute to enhance and augment the reality and interactivity of the next-generation 3-D image communication, which we call "augmented spatial communication".

2. SYNTHESIZING VIRTUAL VIEWS

In this section, the authors review the basic concept of multi-view images and propose the method of view interpolation in the light ray data space.

2.1. Basics of Multi-View Images

Fig.1 illustrates a top view of a camera array system, in which four cameras are aligned horizontally.

P coordinates denote the positions of the cameras on a horizontal line in a 3-D space, and x, the positions of pixels on a horizontal line in each image. The Px plane as shown in Fig.1 is called an EPI (Epipolar Plane Image).[15] The followings are the important characteristics of EPIs.

- All the pixels, corresponding to a point in the space, are aligned on a straight line in an EPI.
- The inclination of the straight line depends on the distance between the camera array and the point in the space.

When tremendous numbers of cameras are densely aligned, continuous structure will appear in the Px plane as shown in the bottom of Fig.2. When cameras are aligned vertically, we can also define the Qy plane, where Q coordinates denote the positions of cameras on a vertical line in the space, and y, the positions of pixels on a vertical line in each image.

Consider the case that X-axis denotes the horizontal axis in the space, Y-axis, the vertical one. We can define the plane, on which cameras are arranged, as $Z = 0$ (See Fig.2).

In this case, the straight lines, corresponding to the point (X, Y, Z) in the space, can be represented as follows[8] :

$$\begin{aligned} P &= X + Zx \\ Q &= Y + Zy \end{aligned} \tag{1}$$

where, θ is defined as shown in Fig.2, and $x = \tan\theta$.

From Eq.(1), we can explain the followings.

- In the Px plane, a line $P = const.$ corresponds to a horizontal line in the image captured on the plane $Z = 0$ (See Fig.1). x denotes the directions of light rays, which pass through the focal points of the cameras.
- In the Px plane, a line $P = ax + b$ $(a < 0)$ corresponds to an object point in the region $Z < 0$ (See the bottom of Fig.2). The process of structure recovery can be regarded as the process of decomposing the Px plane into a set of lines $P = ax + b,\ (a < 0)$.
- In the Px plane, a line $P = ax + b$ $(a > 0)$ corresponds to a viewpoint in the region $Z > 0$ (See Fig.2). By gathering the light ray data on the line, we can synthesize a virtual view corresponding to the viewpoint.

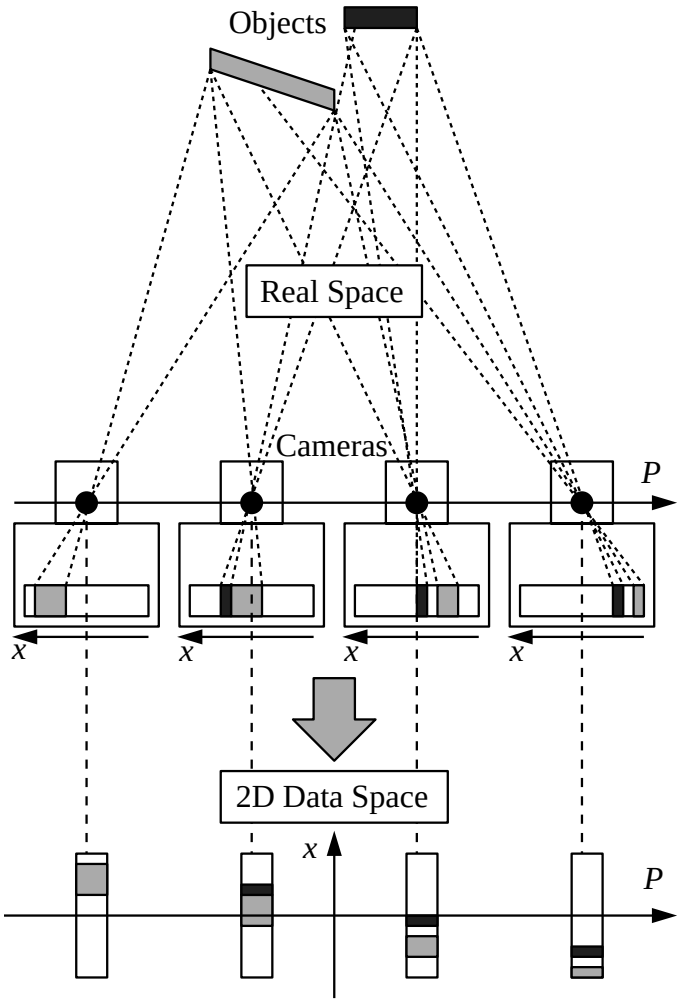


Figure 1. Real space to data space.

The Qy plane has the same characteristics as the Px plane.

Since the above discussion has been concentrated on a line in an image, we can just consider the 2-D data spaces Px and Qy plane. In order to extend the discussion to a 2-D image, we must consider the 4-D data space $PQxy$. In this paper, we represent the values of light ray data stored in the 4-D data space as $f(P, Q, x, y)$.

Consequently, by considering a 4-D data space $f(P, Q, x, y)$, we can synthesize an image $I_{XYZ}(x, y)$, whose viewpoint is (X, Y, Z) in the 3-D real space, as follows :

$$I_{XYZ}(x, y) = f(P, Q, x, y)|_{P=X+Zx, Q=Y+Zy}. \tag{2}$$

We can see that the images captured on a plane $Z = 0$ can be represented as

$$I_{XY0}(x, y) = f(X, Y, x, y). \tag{3}$$

This leads to that Eq.(2) allows us to synthesize a virtual view, whose viewpoint is not on the plane $Z = 0$, from a set of images captured on the plane $Z = 0$.

2.2. Interpolating 4-D Data Space

It is very difficult to arrange CCD cameras densely enough to produce a dense 4-D data space. Consequently, a sparsely sampled data space as shown in Fig.1 will be obtained from a camera array system. So, it is very important to fill up the data space in real-time.

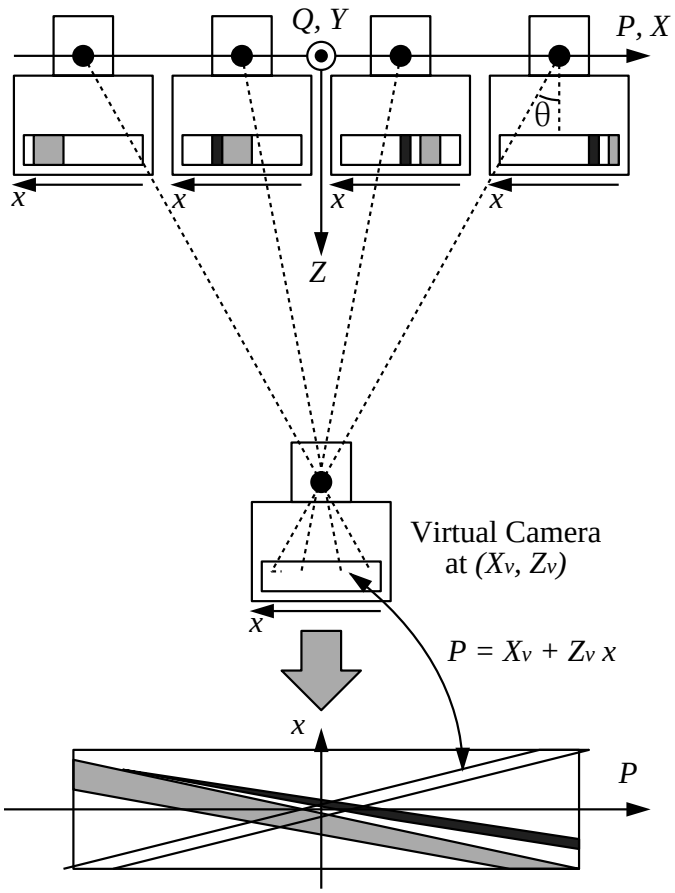


Figure 2. How to synthesize a virtual camera's view.

Let K denote the number of cameras. We can represent the positions of cameras as $(P_k, Q_k) \quad (k = 0, \ldots K-1)$. Let $f_R(P_k, Q_k, x, y)$ denote the data space sampled by the cameras. Fig.1 can be regarded as an example of $f_R(P_k, Q_k, x, y)$. The aim of this section is to synthesize a densely sampled data space $f(P, Q, x, y)$ from a sparsely sampled $f_R(P_k, Q_k, x, y)$. We call this process "light ray interpolation".

Firstly, a simple method A is formulated as follows :

$$f(P, Q, x, y) = f_R([P], [Q], x, y) \tag{4}$$

where

$$\begin{aligned} [P] &= P_{k_o} \\ [Q] &= Q_{k_o} \\ (P - P_{k_o})^2 + (Q - Q_{k_o})^2 &= \min_k \left\{ (P - P_k)^2 + (Q - Q_k)^2 \right\}. \end{aligned} \tag{5}$$

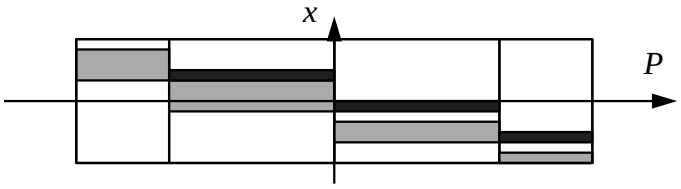
Fig.3(a) shows an example of the results of the method A. We can see discontinuities in the data space.

In order to suppress the discontinuities, the following method B can be formulated.

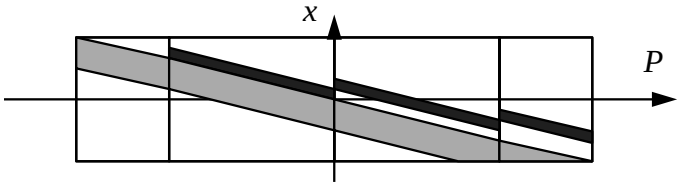
$$f(P, Q, x, y) = f_R([P], [Q], \langle x \rangle, \langle y \rangle) \tag{6}$$

where,

$$\begin{aligned} \langle x \rangle &= x - \frac{P - [P]}{Z_a} \\ \langle y \rangle &= y - \frac{Q - [Q]}{Z_a} \end{aligned} \tag{7}$$



(a) $Z = -\infty$



(b) $Z = Z_a(const.)$

Figure 3. Interpolation of a sparsely sampled data space.

Fig.3(b) shows an example of the result of the method B. In this method, all the scene objects are assumed to be placed on a plane $Z = Z_a$ ($Z_a < 0$). From Eq.(7), we can regard the method A as a special case of the method B, in which the objects are assumed to be placed on $Z_a = -\infty$.

It is sure that even the method B cannot avoid the discontinuities in the interpolated data space. More complex methods can be formulated to avoid this problem.[7,12–14] In this paper, however, for the purpose of real-time interpolation, the authors apply the method B to our system, and allows observers to adjust the interpolation parameter Z_a, interactively.

3. REAL-TIME INTERACTION WITH 3-D SCENES IN MOTION

In this section, the authors propose a real-time method of rendering arbitrary views of 3-D scenes in motion.

3.1. System Configuration

In order to obtain the data space $f(P, Q, x, y)$ in real-time, multiple views of a 3-D scene must be captured by a computer simultaneously. As the number of cameras increases, the capturing process gets heavy and difficult. In order to suppress the load of the capturing process, multiple images are combined into one image at the cost of resolution, in this paper.

Fig.4 shows the configuration of our system. We have sixteen cameras and five quad processors (QP). Four video sequences are combined into one sequence by a quad processor. Firstly, four quad processors combine sixteen video sequences into four sequences. Then, the combined sequences are integrated into one sequence by the fifth quad processor. Thus, sixteen video sequences are captured by a computer simultaneously.

The input video sequence can be regarded as the sparsely sampled data space $f_R(P_k, Q_k, x, y)$. The computer renders virtual views $I_{XYZ}(x, y)$ according to the position of observer's eye (X, Y, Z) and the interpolation parameter (focus point) Z_a. This rendering process, accompanied by the interpolation process, is so heavy that it is difficult to realize a real-time system without a hardware graphics accelerator.

3.2. Processing on Video Texture

For the purpose of real-time processing, we adopt a hardware graphics accelerator, by which a video sequence is treated as a video texture data. In this case, the 4-D data space $f_R(P_k, Q_k, x, y)$ is represented as a 2-D texture data $T(t_x, t_y)$ $(0 \leq t_x < 1, 0 \leq t_y < 1)$.

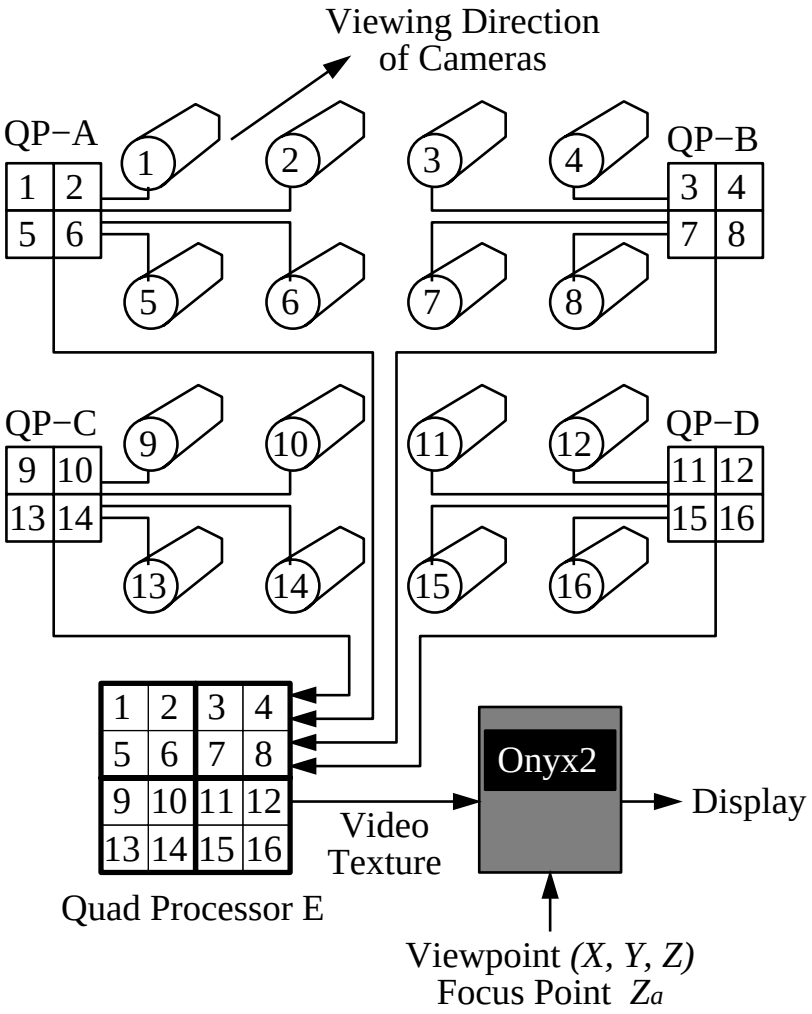


Figure 4. System configuration.

In this paper, we just consider the case where the cameras are arranged on a lattice shown in Fig. 4. In this case, the position of the k-th camera, (P_k, Q_k), can be represented as follows :

$$\begin{aligned} k &= n + mN \\ P_k &= nD \\ Q_k &= mD \end{aligned} \tag{8}$$

where,

N : number of cameras aligned horizontally
M : number of cameras aligned vertically
K : number of cameras $(N \times M)$
D : distance between cameras

$$\begin{aligned} k &= 0, 1, \cdots, K-1 \\ n &= 0, 1, \cdots, N-1 \\ m &= 0, 1, \cdots, M-1. \end{aligned}$$

Then, the following relationship can be derived from Eqs.(5) and (8).

$$\begin{aligned} [P] &= n_o D \\ [Q] &= m_o D \end{aligned} \tag{9}$$

where,

$$\begin{aligned} \mid P - n_o D \mid &= \min_n \mid P - nD \mid \\ \mid Q - m_o D \mid &= \min_m \mid Q - mD \mid. \end{aligned} \tag{10}$$

Ideally, the image I_k, captured by the k-th camera ($k = n + mN$), will appear in the following region of 2-D texture data $T(t_x, t_y)$.

$$\begin{aligned} \frac{n}{N} \leq \quad t_x \quad < \frac{n+1}{N} \\ \frac{m}{M} \leq \quad t_y \quad < \frac{m+1}{M} \end{aligned} \tag{11}$$

So, the texture coordinate (t_x, t_y) corresponding to the pixel data $I_k(x, y)$ can be represented as follows :

$$\begin{aligned} t_x &= \{\ n\ +\ \frac{x}{x_l}\ \}\ \frac{1}{N} \\ t_y &= \{\ m\ +\ \frac{y}{y_l}\ \}\ \frac{1}{M} \end{aligned} \tag{12}$$

where,

$$\begin{aligned} x_l &: \text{horizontal size of images } (0 \leq x < x_l) \\ y_l &: \text{vertical size of images } (0 \leq y < y_l). \end{aligned}$$

Eq.(12) represents the relationship between the sparsely sampled data space $f_R(P_k, Q_k, x, y)$ ($k = n + mN$) and the texture data $T(t_x, t_y)$. Furthermore, from Eqs. (6), (7), (9), (10) and (12), the relationship between the continuous data space $f(P, Q, x, y)$ and $T(t_x, t_y)$ can be formulated as follows :

$$\begin{aligned} t_x &= \{\ n_o\ +\ \frac{\langle x \rangle}{x_l}\ \}\ \frac{1}{N} = \{\ n_o\ +\ \frac{x}{x_l}\ -\ \frac{P - n_o D}{x_l Z_a}\ \}\ \frac{1}{N} \\ t_y &= \{\ m_o\ +\ \frac{\langle y \rangle}{y_l}\ \}\ \frac{1}{M} = \{\ m_o\ +\ \frac{y}{y_l}\ -\ \frac{Q - m_o D}{y_l Z_a}\ \}\ \frac{1}{M}. \end{aligned} \tag{13}$$

We can synthesize a virtual view $I_{XYZ}(x, y)$, whose viewpoint is (X, Y, Z), directly from the video texture $T(t_x, t_y)$ by applying Eqs.(2), (10) and (13). This simplicity enables us to realize a real-time interpolation and rendering.

3.3. Suppressing Differences between Cameras

Eq.(13) is designed just for the ideal cases. It is, however, not practical to align the directions of optical axes of all cameras, precisely. This is because the directions are slightly different from each other, even if the cameras are the same products.

In this paper, in order to extend the applicability of the method, scaling factors α_x, α_y and translations $o_x(k), o_y(k)$ are introduced to Eq.(13) as follows :

$$\begin{aligned} t'_x &= \alpha_x\ t_x + o_x(k) \\ t'_y &= \alpha_y\ t_y + o_y(k). \end{aligned} \tag{14}$$

The translation parameters $o_x(k)$ and $o_y(k)$, given to each camera, are utilized to virtually align the directions of optical axes. Theoretically, translation is not enough for this purpose, but expected to work well when the difference of the directions between cameras is small. The scaling factors α_x and α_y are for the quad processors.

4. EXPERIMENTAL RESULTS

Fig.5 shows the camera array system, on which sixteen CCD cameras are arranged. Each camera is four centimeters apart from the neighboring one. The sixteen video sequences are combined into one sequence by five quad processors.

Fig.6(a) shows an example of the combined image. White lines illustrate how the corresponding points of the sixteen views appear irregularly on the combined image. In order to arrange the corresponding points regularly, we can translate each view as shown in Fig.6(b). The directions and the distances of the translations are calculated by the least square method, and are utilized as the $o_x(k)$ and $o_y(k)$ values in Eq.(14). This calibration is applied for every frame in real-time.

Figs.7, 8 and 9 show the combinations of input video sequence, and the result of real-time video synthesis. The virtual views (bottom of the Figs.) are synthesized from the sets of real views (top of the Figs.). We can see how the real views are warped and stitched in the virtual views. In this experiment, the interpolation parameter Z_a is set to suppress the discontinuities on the region of man.

In this sequence, the viewing position is virtually swinging back and forth, while the camera array is physically fixed. There are two walls in front of sculptures and a man. We can see how the occlusion occurs between the walls and the sculptures. This visual effect is very three-dimensional, because we cannot reproduce such effect by scaling a two-dimensional image.

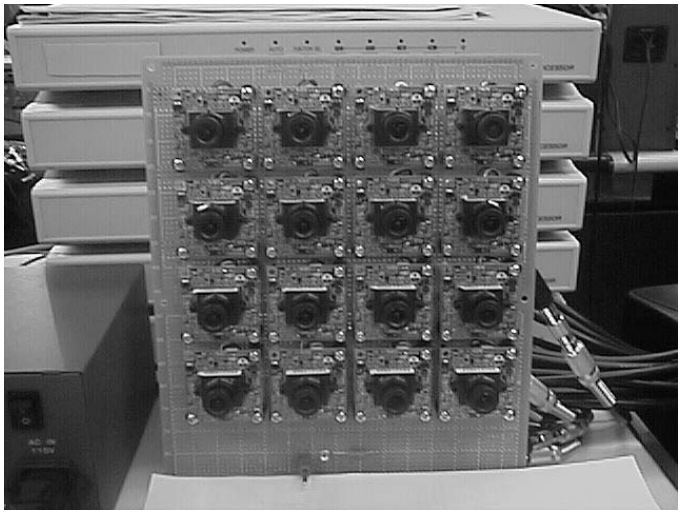


Figure 5. Camera array system.

5. CONCLUSIONS

The authors have proposed a real-time method of synthesizing arbitrary views of a real scene in motion. A camera array system, which consists of sixteen cameras, is constructed to take multi-view video sequences. Since the characteristics of the cameras (especially the directions of optical axes) are slightly different from each other, the authors realized a real-time method, which can virtually adjust the directions of optical axes by translating images on CCD planes. From a set of adjusted video sequences of a scene in motion, the proposed method can synthesize virtual views of the scene corresponding to the viewpoints of observers, in real-time. While the cameras are fixed on a plane, observers can virtually move their viewpoints freely in a 3-D space. So, we can regard the method as a kind of discrete holographic video.

There is much future work to be done on this topic. It will be important to suppress or blur the discontinuities on the synthesized views. Future research also includes how to estimate the optimum value for the interpolation parameter Z_a. With these extensions, the authors believe that the real-time video-based rendering will contribute to enhance and augment the reality and interactivity of the next-generation 3-D image communication, which we call "augmented spatial communication".

(a) Original data.

(b) Adjusted data.

Figure 6. Result of texture calibration.

REFERENCES

1. Y. Kajiki, H. Yoshikawa, and T. Honda, "Three-dimentional display with focused light array," in *SPIE Proc. Practical Holography X*, vol. 2652, pp. 106 – 116, 1996.
2. N. Greene, "Environment mapping and other applications of world projections," *IEEE Computer Graphics and Applications* **6**(11), pp. 21 – 29, 1986.
3. S. E. Chen, "QuickTimeVR - an image-based approach to virtual environment navigation," in *ACM SIGGRAPH'95*, pp. 29 – 38, 1995.
4. S. E. Chen and L. Williams, "View interpolation for image synthesis," in *ACM SIGGRAPH'93*, pp. 279 – 288, 1993.

(a) Adjusted texture data.

(b) Synthesized virtual view.

Figure 7. Result of real-time video-based rendering (1).

5. L. McMillan and G. Bishop, "Plenoptic modeling : An image-based rendering system," in *ACM SIGGRAPH'95*, pp. 39 – 46, 1995.
6. T. Kanade, P. Rander, and P. J. Narayanan, "Virtualized reality : Constructing virtual worlds from real scenes," *IEEE Multimedia* **4**(1), pp. 34 – 47, 1997.
7. T. Naemura and H. Harashima, "Fractal coding of a multi-view 3-D image," in *IEEE Intern. Conf. on Image Process. '94*, vol. III, pp. 107 – 111, 1994.
8. T. Fujii, *A Basic Study on the Integrated 3-D Visual Communication.* PhD thesis, Dept. of Elec. Eng., The Univ. of Tokyo, 1994. (in Japanese).

(a) Adjusted texture data.

(b) Synthesized virtual view.

Figure 8. Result of real-time video-based rendering (2).

9. A. Katayama, K. Tanaka, T. Oshino, and H. Tamura, "Viewpoint-dependent stereoscopic display using interpolation of multiviewpoint images," in *SPIE Proc. Stereoscopic Displays and Virtual Reality Systems II*, vol. 2409, pp. 11 – 20, 1995.
10. T. Yanagisawa, T. Naemura, M. Kaneko, and H. Harashima, "Handling of 3-dimensional objects in ray space," in *Inform. and Syst. Society Conf. of IEICE*, D-169, 1995. (in Japanese).
11. M. Levoy and P. Hanrahan, "Light field rendering," in *ACM SIGGRAPH'96*, pp. 31 – 42, 1996.
12. S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph," in *ACM SIGGRAPH'96*, pp. 43 – 54, 1996.

(a) Adjusted texture data.

(b) Synthesized virtual view.

Figure 9. Result of real-time video-based rendering (3).

13. T. Naemura, *Ray Based Coding of Real Space and Its Application to Augmented Spatial Communication*. PhD thesis, Dept. of Elec. Eng., The Univ. of Tokyo, 1996. (in Japanese).
14. T. Naemura, M. Kaneko, and H. Harashima, "3-D visual data compression based on ray-space projection," in *SPIE Proc. Visual Commun. and Image Process. '97*, vol. 3024, pp. 413 – 424, 1997.
15. R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis : An approach to determining structure from motion," *Computer Vision* **1**, pp. 7 – 55, 1987.