Ray-based approach to Integrated 3D Visual Communication

Takeshi Naemura, Hiroshi Harashima

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

ABSTRACT

For a high sense of reality in the next-generation communications, it is very important to realize three-dimensional (3D) spatial media, instead of existing 2D image media. In order to comprehensively deal with a variety of 3D visual data formats, the authors first introduce the concept of "Integrated 3D Visual Communication," which reflects the necessity of developing a neutral representation method independent of input/output systems. Then, the following discussions are concentrated on the ray-based approach to this concept, in which any visual sensation is considered to be derived from a set of light rays. This approach is a simple and straightforward to the problem of how to represent 3D space, which is an issue shared by various fields including 3D image communications, computer graphics, and virtual reality. This paper mainly presents the several developments in this approach, including some efficient methods of representing ray data, a real-time video-based rendering system, an interactive rendering system based on the integral photography, a concept of virtual object surface for the compression of tremendous amount of data, and a light ray capturing system using a telecentric lens. Experimental results demonstrate the effectiveness of the proposed techniques.

Keywords: Integrated 3D Visual Communication, light field, ray space, image-based rendering, video-based rendering, integral photography, virtual object surface, data compression, telecentric lens, orthogonal projection.

1. INTRODUCTION

Although existing 2D image media may sometimes invoke sensations like "I feel like I'm in another space," or "I instinctively stretched out my hand to an object in front of me," they cannot be said to have sufficient power of expression. Further advancements in virtual reality technologies and 3D image communications technologies are therefore needed to achieve a high sense of presence and reality in the next-generation communications and information environment.

At present, 3D display technology is at the stage where a variety of systems are being studied, and the coming years should see amazing developments through advanced research. Under these circumstances, however, specialized 3D information environments may be constructed for specific display systems leading to incompatibility in sending, receiving, and sharing 3D data in the future. To prevent this problem from occurring, a flexible 3D-information environment that can support

Correspondence : Email: naemura@hc.t.u-tokyo.ac.jp; WWW: http://www.hc.t.u-tokyo.ac.jp/~naemura

a variety of display systems must be constructed. Along this line of thinking, the authors have already proposed the concept of "Integrated 3D Visual Communication."[1] This concept reflects the necessity of developing a neutral representation method (encoding method) independent of input/output systems (i.e., camera and display systems, referred to below as "3D image formats") like stereo, multi-view, and hologram, as shown in Fig. 1. Such a method deals with data directly associated with 3D space in contrast to those making up individual 3D image formats, and we refer to it as "spatial representation" in the following discussion.

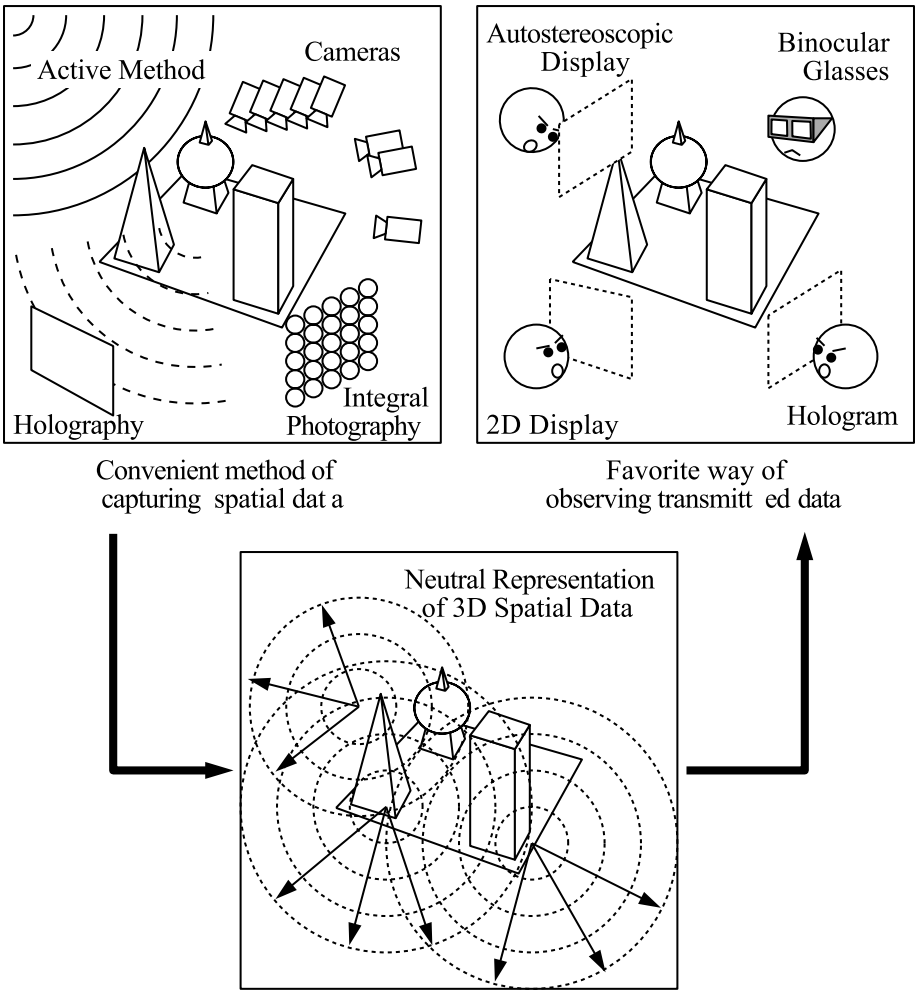


Figure 1. Basic concept of "Integrated 3-D Visual Communication."

Up to now, research into 3D image-encoding technology has mainly targeted individual 3D image formats.[2,3] Here, methods that use structural models of objects that are conscious of spatial representation have also been studied.[4–8] Representation by structural models is highly compatible with computer graphics (CG) and is expected to be especially applicable to virtual reality (VR) applications. However,

to appropriately estimate a structural model of an object from a 3D image, a wide variety of problems that have turned up in the field of computer vision (CV) must be solved, and this cannot be easily achieved at present.

On the other hand, a technique using ray data has been proposed as a means of representing visual data in 3D space.[9–14] This technique processes sets of ray data in contrast to conventional techniques that target image data, and is expected to achieve a high degree of freedom in processing by targeting ray data. It can be treated, moreover, as an especially useful concept in formulating a method of spatial representation that can comprehensively deal with a variety of 3D image formats.

In this paper, we first investigate an efficient method of representing ray data.[15] Here, by ignoring changes that accompany the propagation of light (such as interference and attenuation), we decrease the number of dimensions of data space for representing ray data and formalize a specific representation technique. We then show that spatial representation by this technique is equivalent to a method of representation based on sets of orthogonal (parallel) projection images independent of viewpoint. After this, we describe the construction of a system that can perform all processing from input to output in real time.[16] This system is a landmark development not only as a ray-based approach to representing visual data but also as an image-based rendering system. We then describe the application of this ray-based representation method to integral photography, a type of 3D display technology, and show how its usefulness has been confirmed by experiment.[17] Next, with respect to the huge amount of data used in ray-based representation, we investigate how to compress this data efficiently and introduce the concept of a virtual object surface.[18] Finally, we report on a method for directly acquiring ray data using a telecentric lens.[19]

2. SPATIAL REPRESENTATION BY RAY DATA

"Visual data" means information on light rays which propagate through space. "Transmitting all information on light rays which fill 3D space" therefore makes possible the "visual transmission of the 3D space." With this in mind, we here outline a method of spatial representation focused on ray data.[15]

2.1. Representing 3D spatial information by ray data

The set of all light rays that pass through point (X, Y, Z) in 3D space is called a "pencil." Three-dimensional space is filled with innumerable pencils, and the individual rays that make up a pencil are distinguished by direction (θ, ϕ). Accordingly, we let f denote ray data (brightness, color, etc.) in 3D space and define f as the following function consisting of five parameters.[20]

$$\text{ray data} = f(X, Y, Z, \theta, \phi) \tag{1}$$

In other words, five dimensions of data space are needed to record ray data.

Based on this function, pencil data at viewpoint $A(X_a, Y_a, Z_a)$ (all surrounding visual information in a QuickTimeVR panorama image[21]) can be written as follows.

$$f_a(\theta, \phi) = f(X_a, Y_a, Z_a, \theta, \phi) \tag{2}$$

Capturing an image by a pinhole camera can be treated as an operation that obtains pencil data as image data. Conversely, given an observer at viewpoint A in a display system, stored pencil data may be read out and projected onto a picture screen. In this manner, capturing an image means the "sampling of pencil data," while image synthesis is achieved by the "reading out of pencil data." Also, by ignoring phase information of light, it becomes possible to not only synthesize pinhole images, but to also synthesize images virtually from $f(X, Y, Z, \theta, \phi)$ at various focal distances by supposing a lens system.[22] In short, image data corresponding to any viewpoint, viewing direction, and focal distance are embedded in $f(X, Y, Z, \theta, \phi)$.

2.2. Spatial representation for restricted viewpoints

When processing data obtained by moving a camera in the horizontal direction (along the line $(Y, Z) = (Y_a, Z_a)$, which is parallel to the X-axis), such as in stereo images or multi-view images, the f function takes on the following form.

$$f_a(X, \theta, \phi) = f(X, Y_a, Z_a, \theta, \phi) \tag{3}$$

In more general terms, an image sequence obtained by moving a camera along a straight line or curve $(X, Y, Z) = C(p)$ (including, for example, an image sequence of a walkthrough application) becomes as follows.

$$f_C(p, \theta, \phi) = f(X, Y, Z, \theta, \phi)|_{(X,Y,Z)=C(p)} \tag{4}$$

Likewise, an image sequence obtained on a curved surface $(X, Y, Z) = S(p, q)$ can be written as follows.

$$f_S(p, q, \theta, \phi) = f(X, Y, Z, \theta, \phi)|_{(X,Y,Z)=S(p,q)} \tag{5}$$

Accordingly, when camera position (observer viewpoint) is restricted, the above technique decreases the number of parameters and represents spatial data in a more efficient manner.

2.3. Spatial representation for viewpoints with greater degrees of freedom

We consider the case in which boundary surface S divides a 3D space into two spaces. One is being described (described zone) and another is the space in which the observer is present (visual zone), as shown in Fig. 2. Here, assuming that "light travels in a straight line within the 3D space without being affected by interference, attenuation, and the like," an image for viewpoint situated at point O in the figure can be virtually synthesized by collecting ray data that passes through S and arrives at O. In other words, establishing the above assumption means that, for the ray data of Eq. (5), the observer can move to positions away from surface S and a 3D degree of freedom can be obtained with respect to the viewpoint.

In this situation, where the viewpoint takes on a 3D degree of freedom, the range of movement is nevertheless limited to either one of the two zones divided by boundary surface S. To put it another way, we can consider two ways of using Eq. (5) depending on which of the two zones is specified as the described zone, as depicted in Fig. 3. Figure 3(a) corresponds to object representation when objects

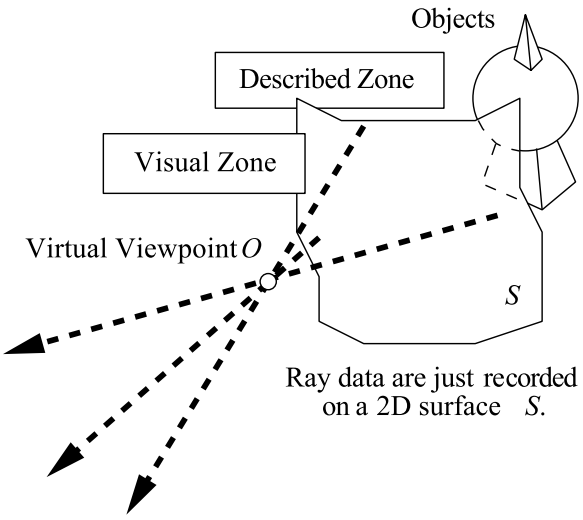
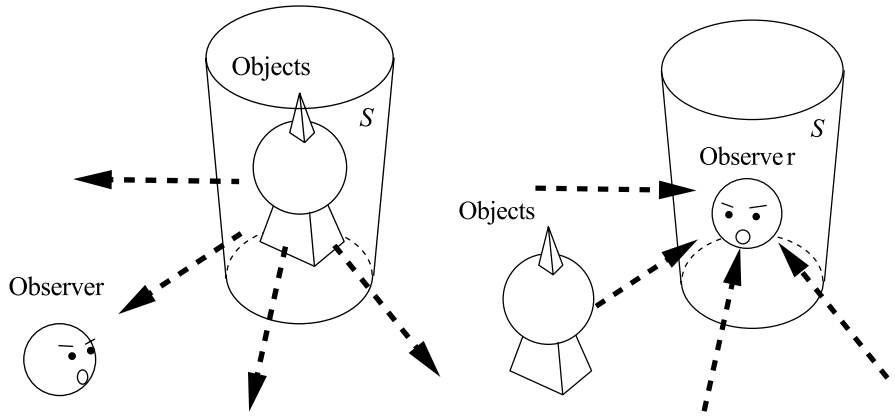


Figure 2. Synthesis of virtual viewpoint images from light rays recorded on a surface.



(a) Application to multiplex hologram and so on.

(b) Application to the CAVE system and so on.

Figure 3. Viewpoint-independent representation of 3-D spatial data.

enclosed in a zone are observed from the outside, and Fig. 3(b) corresponds to environment representation when an observer, enclosed in a zone, observes the outside environment.

Accordingly, having four parameters as in Eq. (5) enables spatial representation with a sufficiently high degree of freedom. Compared to using five parameters, we can see that decreasing the number of dimensions of data space by one results in efficient spatial representation.

We emphasize here that Eq. (5) corresponds to spatial representation dependent

on the shape of surface S. For this reason, a conversion process will be required when camera arrangement S_i at the time of image capturing differs from display shape S_o at the time of display. In short, conversion techniques dependent on both input and output must be defined exactly for the number of such combinations. This property detracts from the flexibility of a 3D-information environment.

This problem can be solved, however, by neutral representation of 3D space independent of the boundary surface's shape. Here, by preparing a conversion from the input system to a neutral method of representation and a conversion from this neutral method of representation to the output system, any combination of input/output can be handled in a flexible manner. The following discusses a technique for achieving such neutrality independent of boundary-surface shape for the efficient four-parameter spatial representation method described above.

2.4. Spatial representation independent of boundary-surface shape

With reference to Eq. (5), we denote four-parameter spatial representation independent of boundary-surface shape as follows.

$$f'(P, Q, \theta, \phi) = f(X, Y, Z, \theta, \phi)|_{(X,Y,Z)=S'(P,Q)} \tag{6}$$

In this case, data of light propagating in a fixed direction (θ_a, ϕ_a) can be written as follows.

$$f'_a(P, Q) = f'(P, Q, \theta_a, \phi_a) \tag{7}$$

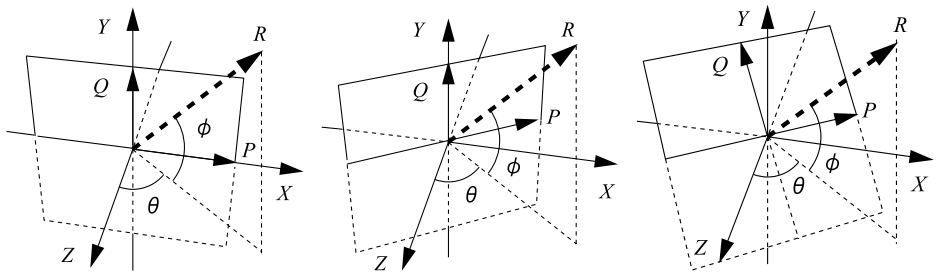
Equation (7) corresponds to the recording of data for light rays oriented in the same direction, that is, data corresponding to an orthogonal projection of space, and the index of this data is given by the PQ coordinate system. In Eq. (5), as well, the pq coordinate system can be an index of orthogonal-projection data, but this system would be dependent on the shape of surface S.

With the above in mind, the purpose of this section is to define an easy to handle PQ coordinate system as an index to orthogonal-projection data. Differences in surface shape can then be absorbed by coordinate conversion between the pq coordinate system and PQ coordinate system.

Here, we define axis R coinciding with the direction of a light ray (optical axis) and denote its unit vector in the XYZ coordinate system as $\boldsymbol{r}$. Now, given two unit vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ linear independent of $\boldsymbol{r}$ and their corresponding axes P and Q, coordinate conversion from the XYZ coordinate system to the PQR coordinate system can be formalized. Specifically, vector $\boldsymbol{a} = (X_a, Y_a, Z_a)$ in the XYZ coordinate system converts as shown below to the PQR coordinate system.

$$\boldsymbol{a} = (X_a, Y_a, Z_a) = P\boldsymbol{p} + Q\boldsymbol{q} + R\boldsymbol{r} \tag{8}$$

Because changes to light during propagation are recorded along the R-axis, ignoring these changes would make values of the R coordinate meaningless. The PQ coordinate system, moreover, is an index of data for space projected in the $\boldsymbol{r}$ direction. The following investigates specific techniques for obtaining $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{r}$.



(a) Planer coordinate system.

(b) Cylindrical coordinate system.

(c) Spherical coordinate system.

Figure 4. Coordinate systems for recording ray data.

2.5. Planar recording of ray data

We first consider the case where the P and Q axes coincide with the X and Y axes, respectively, as shown in Fig. 4(a).

Indicating ray direction as (θ, ϕ), we consider the coordinate systems shown. The $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{r}$ unit vectors can be expressed as follows in terms of the XYZ coordinate system.

$$\begin{aligned} \boldsymbol{p} &= (1,\, 0,\, 0) \\ \boldsymbol{q} &= (0,\, 1,\, 0) \\ \boldsymbol{r} &= (\sin\theta\cos\phi,\, \sin\phi,\, \cos\theta\cos\phi) \end{aligned} \tag{9}$$

Now, substituting these expressions for $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{r}$ in Eq. (8), P and Q can be expressed as follows in terms of (X_a, Y_a, Z_a).

$$\begin{aligned} P &= X_a - Z_a \tan\theta \\ Q &= Y_a - Z_a \tan\phi/\cos\theta \end{aligned} \tag{10}$$

From these equations, the conversion from $f(X, Y, Z, \theta, \phi)$ to $f'(P, Q, \theta, \phi)$ can be defined. In other words, ray-data index (P, Q) can be determined from position coordinates (X_a, Y_a, Z_a) in 3D space. Accordingly, when data on light rays that pass through position $A(X_a, Y_a, Z_a)$ in 3D space are needed, ray data recorded at position (P, Q) computed from Eq. (10) may be read out from $f'(P, Q, \theta, \phi)$.

This technique, however, cannot describe light rays that are parallel to the XY plane, that is, only rays that cross the XY plane can be recorded. For this reason, we call this technique "planar recording of ray data."

2.6. Cylindrical recording of ray data

Next, we define the P, Q, and R axes as the result of rotating the X, Y, and Z axes about the Y-axis by an angle of θ and then rotating only the Z-axis about the

X-axis by an angle of ϕ, as shown in Fig. 4(b). In the same way as discussed above, we can derive the following equations. Here, $\boldsymbol{r}$ is the same as in planar recording.

$$\begin{aligned} \boldsymbol{p} &= (\cos\theta,\, 0,\, -\sin\theta) \\ \boldsymbol{q} &= (0,\, 1,\, 0) \end{aligned} \tag{11}$$

$$\begin{aligned} P &= X_a \cos\theta - Z_a \sin\theta \\ Q &= -X_a \sin\theta \tan\phi + Y_a - Z_a \cos\theta \tan\phi \end{aligned} \tag{12}$$

This technique cannot describe light rays that are parallel to the Y-axis, that is, only rays that cross the cylindrical surface parallel to the Y-axis can be recorded. This technique is therefore called "cylindrical recording of ray data."

2.7. Spherical recording of ray data

Finally, we define the P, Q, and R axes as the result of rotating the X, Y, and Z axes about the Y-axis by an angle of θ and again about the X-axis by an angle of ϕ. In this case, the P, Q, and R axes are always perpendicular to each other. Again, similar to the above discussions, we derive the following equations.

$$\begin{aligned} \boldsymbol{p} &= (\cos\theta,\, 0,\, -\sin\theta) \\ \boldsymbol{q} &= (-\sin\theta\sin\phi,\, \cos\phi,\, -\cos\theta\sin\phi) \end{aligned} \tag{13}$$

$$\begin{aligned} P &= X_a \cos\theta - Z_a \sin\theta \\ Q &= -X_a \sin\theta \sin\phi + Y_a \cos\phi - Z_a \cos\theta \sin\phi \end{aligned} \tag{14}$$

In contrast to the above techniques, this one enables ray data in any direction to be recorded. In other words, all rays that cross the spherical surface defined about the origin can be recorded. We call this technique "spherical recording of ray data."

2.8. Overview of spatial representation system

A spatial representation system based on ray data can handle various input/output systems in a comprehensive manner through the following procedure.

1. Convert 3D image input (e.g., stereo, multi-view, hologram, or polygon formats) to ray data and store in $f'(P, Q, \theta, \phi)$.
2. Interpolate and fill gaps in $f'(P, Q, \theta, \phi)$.
3. Compress and transmit $f'(P, Q, \theta, \phi)$.
4. Read out and display appropriate ray data from $f'(P, Q, \theta, \phi)$ in accordance with the display system.

In general, it is not easy to obtain sufficient ray data by an actual camera system to fill $f'(P,Q,\theta,\phi)$ in a dense manner. In short, simply converting various types of 3D image input to ray data can lead to gaps in $f'(P,Q,\theta,\phi)$. This is why interpolation and synthesis of ray data becomes necessary to fill these gaps. Specifically, virtually interpolated ray data will be added to actual ray data captured by camera so as to generate a densely filled $f'(P,Q,\theta,\phi)$. The significance of this process is that a densely filled $f'(P,Q,\theta,\phi)$ so obtained represents a new world having aspects of both the real and virtual worlds.

The function $f(X,Y,Z,\theta,\phi)$, Eq. (5), and the like can be treated as a set of pencil data $f_a(\theta,\phi)$ equivalent to a perspective projection image. On the other hand, the function $f'(P,Q,\theta,\phi)$ can be treated as the set of PQ planes in an orthogonal projection of space.

3. CONSTRUCTION OF A REAL-TIME SYSTEM

This section describes the constrution of a system that, based on the planar recording method described in section 2.5, performs all processing from input (capture) to output (image synthesis) in real time.[16]

3.1. Transformation of the planar recording method

We consider the case in which the camera's optical axis is parallel to the Z-axis in the XYZ coordinate system. Letting F denote the distance from the camera's focus point to its pick-up surface, the following relationship holds between the position (p_x,p_y) of a pixel on the captured image and the direction (θ,ϕ) of the light ray recorded on that position.

$$\begin{aligned} p_x &= -F\tan\theta \\ p_y &= -F\tan\phi/\cos\theta \end{aligned} \tag{15}$$

Here, we make the following assignments to express ray direction in terms of (x,y) instead of (θ,ϕ).

$$\begin{aligned} x &= p_x/F \\ y &= p_y/F \end{aligned} \tag{16}$$

Furthermore, as the P and Q axes coincide with the X and Y axes, respectively, in the planar recording method, $f'(P,Q,\theta,\phi)$ can be rewritten as follows.

$$\text{ray data} = f'(X,Y,x,y) \tag{17}$$

The following equations can now be derived from Eqs. (10), (15), and (16), and we can see that corresponding points on the object form lines on the Xx and Yy planes (Consider X_a,Y_a,Z_a are constant).

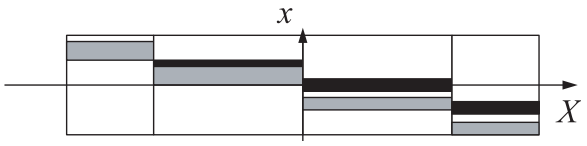
$$\begin{aligned} X &= X_a + Z_a x \\ Y &= Y_a + Z_a y \end{aligned} \tag{18}$$

This is a property conforming to the theory of epipolar plane images.[23]

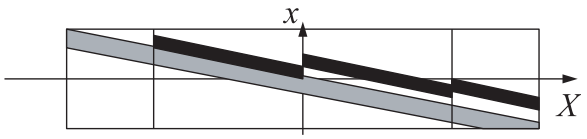
Here, to obtain $f'(X, Y, x, y)$, cameras may be lined up on the XY plane and shots taken. On the other hand, to synthesize image $I_a(x, y)$ at viewpoint (X_a, Y_a, Z_a) from Eqs. (17) and (18), ray data will have to be collected in the following manner.

$$I_a(x, y) = f'(X_a + Z_a x,\ Y_a + Z_a y,\ x,\ y) \tag{19}$$

The purpose of this section is to find a means of performing processing from the acquisition of $f'(X, Y, x, y)$ to the synthesis of $I_a(x, y)$ in real time while varying viewpoint (X_a, Y_a, Z_a) interactively. In general, as a result of restrictions attached to camera spacing, $f'(X, Y, x, y)$ obtained in real time results in discrete sampling in the direction of the X and Y axes in comparison to sampling in the x and y directions. This situation is shown in Fig. 5(a) when seen from above the Xx plane. Here, by approximating the shape of the object at plane $Z = Z_i$, interpolation as shown in Fig. 5(b) can be performed. Also, by creating an interface so that the value of Z_i can be varied interactively, it becomes possible to suppress distortion in the object of attention at the will of the observer.



(a) $Z = \infty$.



(b) $Z = Z_i$(cont.).

Figure 5. Interpolation of a sparsely sampled data space.

3.2. System configuration

Obtaining $f'(X, Y, x, y)$ in real time requires the simultaneous handling of multi-view video input obtained from multiple cameras. In this paper, we adopt the system configuration shown in Fig. 6.

The Quad Processor (QP) in the figure is a piece of equipment that integrates video images from four cameras and outputs a single video image on a screen partitioned into four sections. Consequently, video images from 16 cameras can be integrated into four images each having four sections by using four QP units. Then, by inputting these images into a fifth QP (QP E), the video images from 16 cameras can be integrated into a single image partitioned into 16 sections.

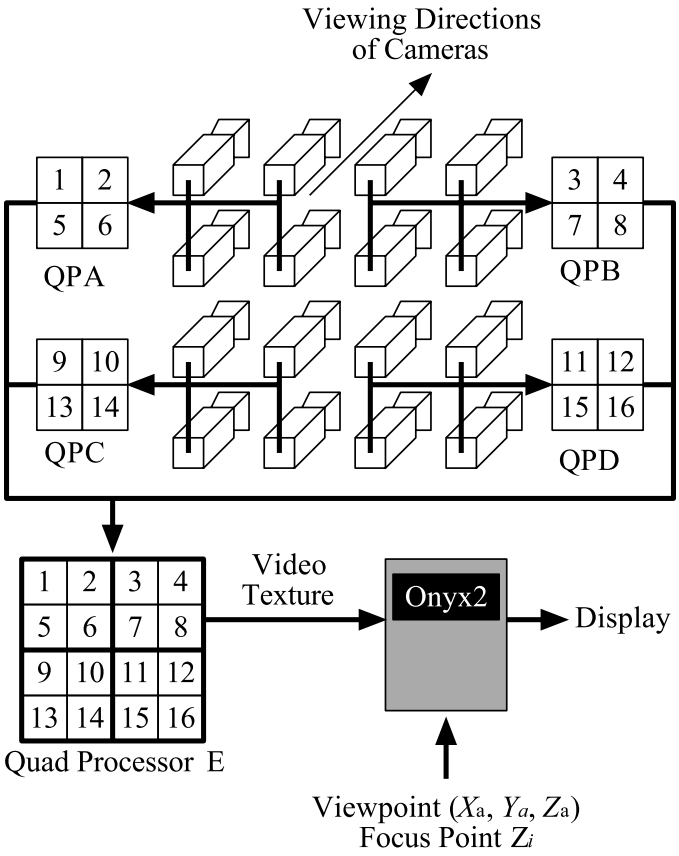


Figure 6. System configuration for the real-time video-based rendering system.

Here, using observer viewpoint (X_a, Y_a, Z_a) and approximate distance Z_i to the subject, the computer processes the 16-section video input and synthesizes desired pictures. In this paper, we handle video input as texture and achieve high-speed processing using a graphic engine.

3.3. Experimental results

An external view of the camera system that we constructed is shown in Fig. 7. It consists of 16 CCD cameras (MTV-2510EM) arranged on a board in a 4-by-4 array with a camera spacing of 4 [cm]. This system can sample, at intervals of 4 [cm], light rays passing through a 12 [cm] square plane.

Examples of moving pictures synthesized in real time are shown in Fig. 8. On the left are 16-section input images, and on the right are synthesized images achieved by a texture-mapping function targeting the areas enclosed by the white squares in the corresponding 16-section input image. These synthesized images are the result of virtually moving the viewpoint back and forth while looking between the walls from the fixed camera system. Specifically, in order from top to bottom in the figure, the camera moves backward (in a direction away from the character figures) in a virtual manner. As shown, the images of these figures become progressively

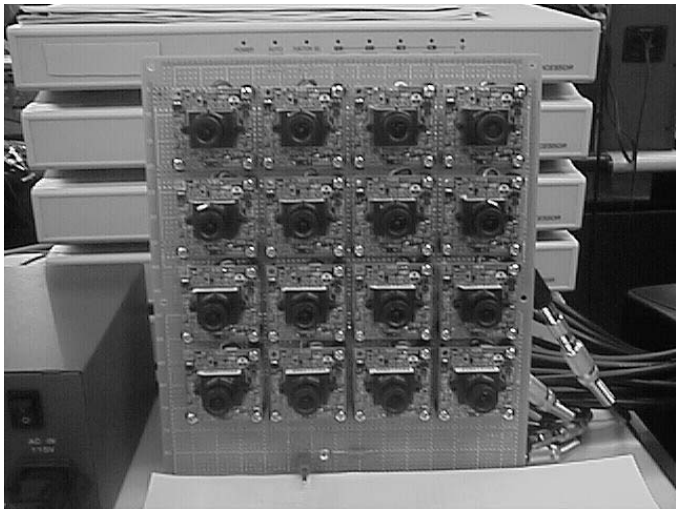


Figure 7. Camera array system.

smaller. Note that occlusion occurring between the wall in the foreground and an inner figure has been reproduced. This is a 3D visual effect that cannot be obtained simply by zooming in or out. Object position Z_i can also be made to coincide with the position of the figures by mouse operation. Here, while errors occur in the foreground wall and background, favorable synthesis is achieved with regard to the area surrounding the figures.

For picture synthesis, we used Onyx2 Reality (R10000 195 MHz $\times$ 2) equipment manufactured by SGI and a video texture function through DIVO (DIgital Video Option). Processing for extracting and aligning texture according to viewpoint was performed at 47 [frames/sec]. This figure is greater than the 30 [frames/sec] of video input, which indicates the possibility of even more complex processing.

4. ACQUISITION OF DENSE DATA SPACE UTILIZING INTEGRAL PHOTOGRAPHY

In section 3, we reported on the real-time application of the planar recording method using a camera array. In this section, we describe a technique that samples ray data at a much greater density than a camera array by using integral photography (IP) as the input format.[17]

4.1. Using a real-time IP system

The NHK Science and Technical Research Laboratories has achieved a real-time full-color IP system using a bundle of optical fibers and a high-definition camera.[24,25] In this system, the high-definition camera picks up the inverted image of a target object as elemental images, the number of which corresponds exactly to the number of lenses (optical fibers). An example is shown in Fig. 9.

Figure 8. Results of real-time video-based rendering.

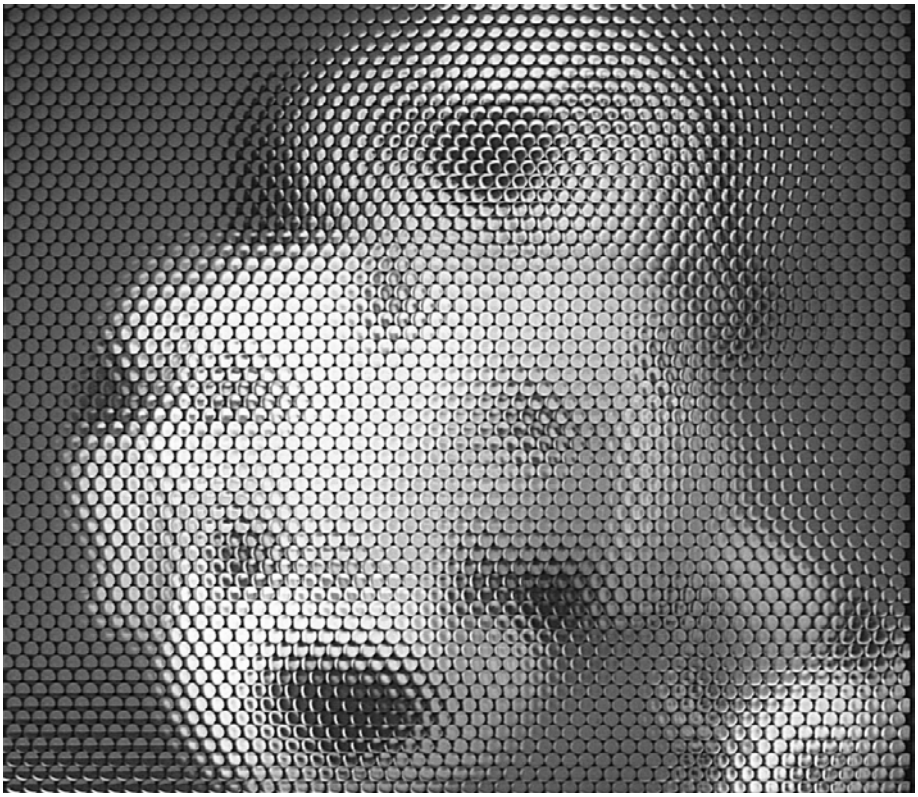


Figure 9. Example of IP image.

In this section, we examine a system that feeds an image like the one shown here to a computer and synthesizes a new image interactively according to the position of the observer without the use of special optics.

The method of section 3 and the method described here are compared in Table 1. In either case, we have a method that deals with function $f'(X, Y, x, y)$ of the planar recording method. There is a difference, however, in how they sample ray data.

Table 1. Comparisons of methods in sections 3 and 4

Method	No. of Views (X, Y resolution)	No. of Pixels in Elemental Image (x, y resolution)
Section 3	4×4	About 180×120
Section 4	54×57	About 20×20

In short, these two input systems differ only in the way they sample ray data, and the algorithm of the ray representation method can be equally applied to both. The experimental results presented below were obtained by applying the algorithm described in section 3 to the IP image (Fig. 9) captured by a high-definition camera.

4.2. Experimental results

Figure 10 shows the results of simulating IP playback optics by computer with respect to the input image of Fig. 9. The number of elemental images in this setup (54×57) corresponds to the resolution of the synthesized image. Figure 11, on the

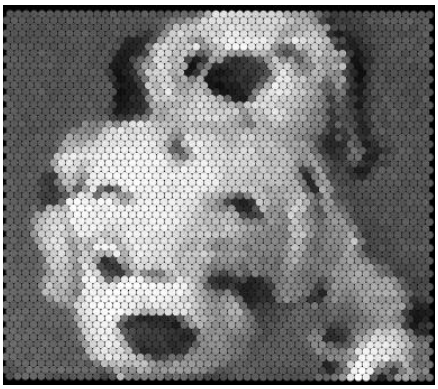


Figure 10. Result of simulating IP playback optics.

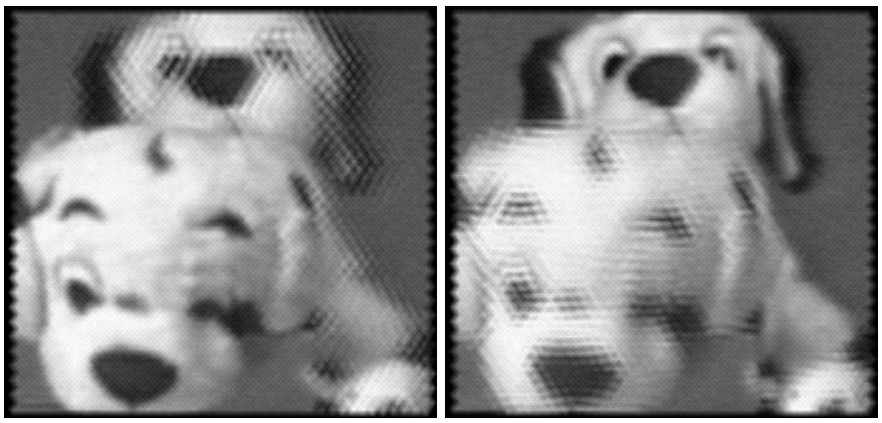


Figure 11. Result of the proposed method.

other hand, shows the results of applying the method of section 3 to IP input. In the left picture, depth parameter Z_i has been adjusted for the dog in the front, while in the right picture, it has been adjusted for the dog in the back. Picture quality partially improves according to object depth. Figure 12 shows results for different types of source pictures. These results demonstrate that the method of section 3 is effective for objects that reflect in a shiny and complex manner and for moving objects like people.

The above images were synthesized in real time while interactively varying viewpoint and depth parameter Z_i of the object.

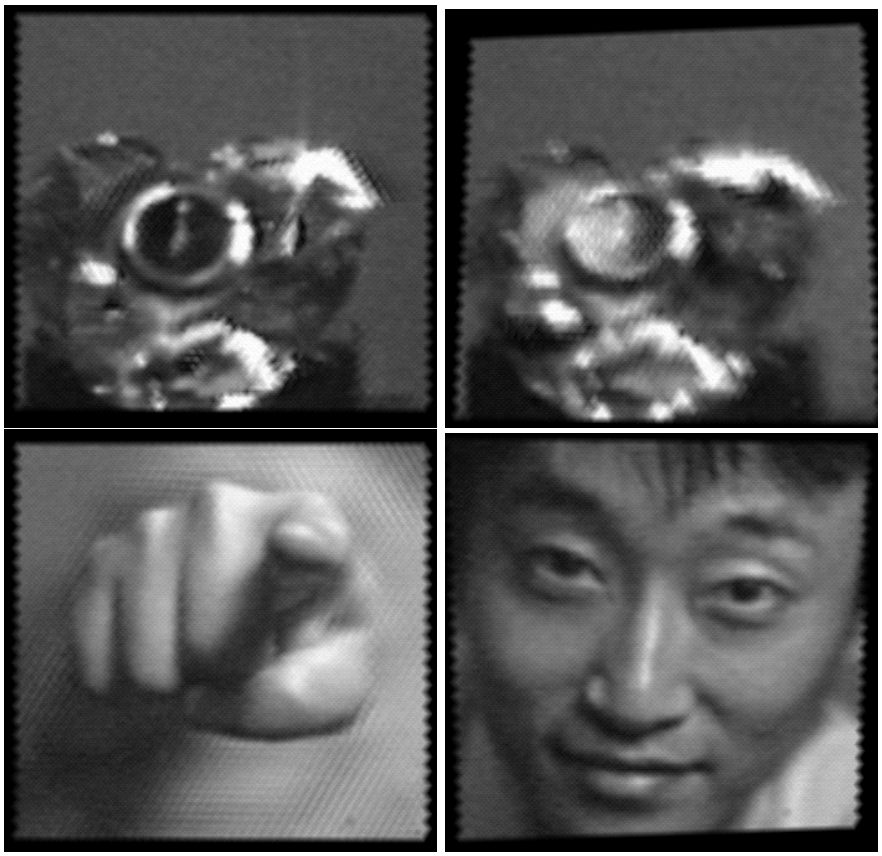


Figure 12. Other results for objects that reflect in a shiny and complex manner and for moving objects like people.

5. COMPRESSION OF RAY DATA

An important issue in the ray representation method is how to efficiently compress the huge volume of spatial data present.

5.1. 4D discrete cosine transform

Compression using the discrete cosine transform (DCT) has found wide use in the field of picture encoding. Here, in contrast to two dimensions in the case of pictures, we can consider four-dimensional DCT targeting the function $f'(P, Q, \theta, \phi)$. In particular, by defining 4D zigzag scanning and the like for DCT coefficients, this approach has been found to be an effective method of compressing ray data.[26]

5.2. Improving compression efficiency by redefining the boundary surface

Let's take another look at the 4D data space $f'(X, Y, x, y)$ described in section 3. We know that data on light rays passing through plane $Z = 0$ will be recorded in $f'(X, Y, x, y)$. If we now consider light rays that pass though plane $Z = Z_a$, we

can derive an expression for 4D data space $g(X, Y, x, y)$ in the same manner as Eq. (19), as shown below.

$$g(X, Y, x, y) = f'(X + Z_a x,\ Y + Z_a y,\ x,\ y) \tag{20}$$

In other words, the boundary surface may be redefined while preserving ray data.

Furthermore, by making use of the fact that redundancy in ray data becomes easier to use by selecting an appropriate value for Z_a, it becomes possible to further improve compression by 4D DCT. In general, a greater improvement in compression efficiency by 4D DCT is observed when setting Z_a near the object and recording ray data in $g(X, Y, x, y)$ as compared to defining a boundary surface far from the object and configuring 4D data space $f'(X, Y, x, y)$.

5.3. Virtual object surface

Expanding on the above ideas, we can do more than just bring the boundary surface closer to the object. We can also define a surface or layer approximating the shape of the object and thereby make it possible to perform even more efficient data compression. This is equivalent to enclosing the object in a 3D display (a surface for which the manner of viewing changes according to viewpoint). In the following, this kind of surface that approximates the shape of the object to handle ray data is called a "virtual object surface."

When recording and transmitting visual information of an object using a virtual object surface, there are two types of data that must be conveyed:

- Data on the shape of the virtual object surface
- Data on the light rays that pass through the virtual object surface

From the standpoint of data compression, we can consider that

- a virtual-object-surface shape that closely approximates the shape of the object results in high correlation among the rays at a single point on the virtual object surface and simplifies the compression of ray data.

On the other hand, we can expect that

- the amount of data needed to describe the shape of a virtual object surface to increase as that shape becomes more complex.

In other words, we have a tradeoff relationship, and there exists an optimal degree of shape approximation that minimizes the total amount of data.[18]

5.4. Experimental results

We compare compression efficiency for ray data under a variety of conditions as described below (Fig. 13).

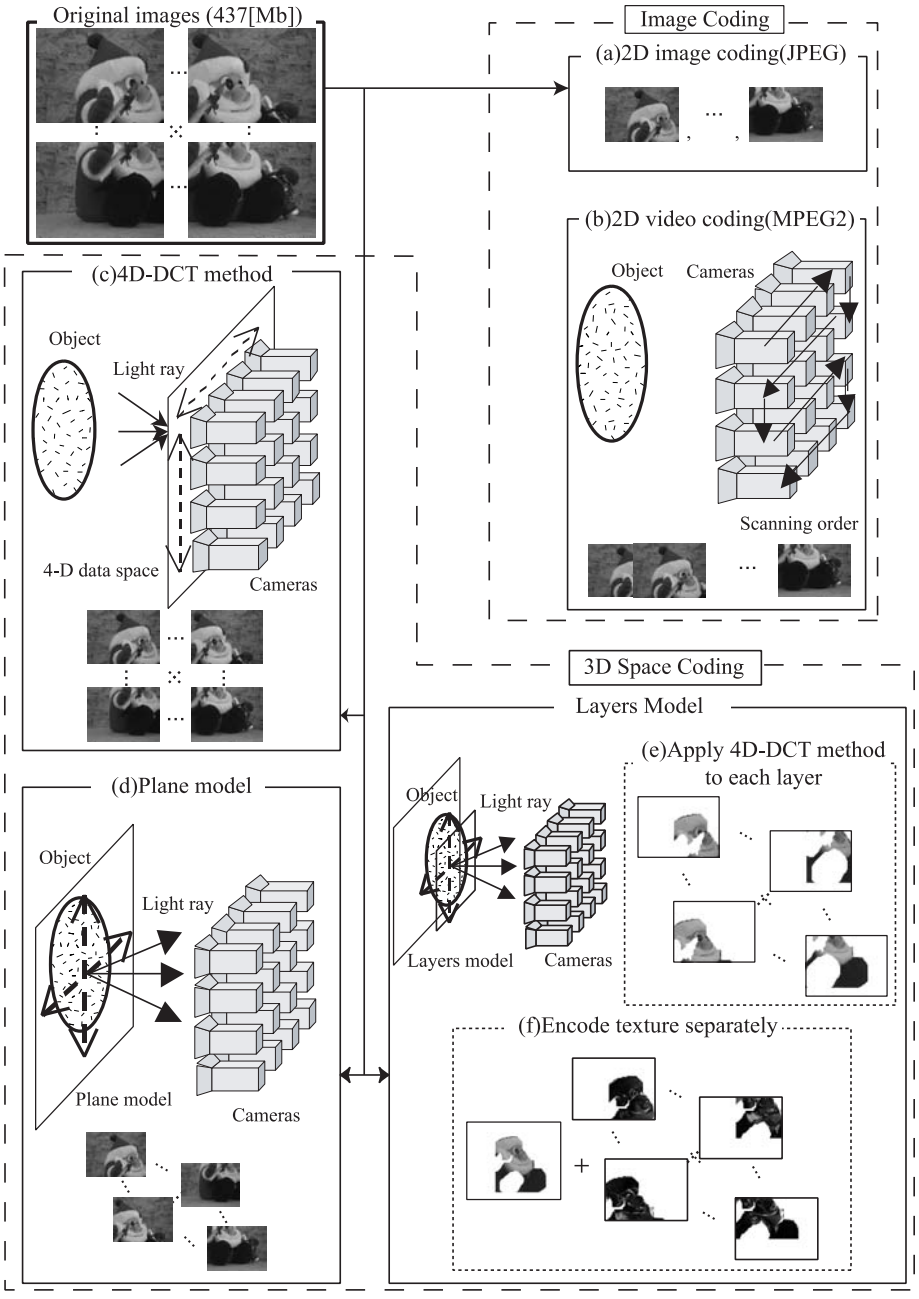


Figure 13. Relationship between various experiments.

(a) Each image of a multi-view picture is compressed by JPEG

(b) A multi-view picture is treated as a moving picture and compressed by MPEG2

(c) Compression with respect to $f'(X, Y, x, y)$ is performed by 4D DCT

(d) Z_a is optimally set and compression with respect to $g(X, Y, x, y)$ is performed by 4D DCT

(e) Each layer in the layered model shown in Fig. 14 is compressed by 4D DCT as a virtual object surface

(f) Texture data in the layered model is treated separately in the technique described in (e)

Figure 14. Example of layered model.

Conditions (a) and (b) constitute an experiment with the aim of comparing conventional techniques with the proposed method. Conditions (c), (d), and (e), on the other hand, constitute an experiment to compare various techniques that use the 4D DCT method described earlier. Condition (f), meanwhile, has been added to these experiments as a technique for separately transmitting texture in the layered model beforehand considering the difficulty of interactive image display if 4D DCT is time consuming.

Experimental results are shown in Fig. 15. The horizontal axis shows total code length required to transmit the multi-view picture, and the vertical axis shows average SNR at various capture points. The amount of data in the multi-view picture before encoding is 437 [Mb].

Using a virtual object surface in technique (d) produces better results than conventional techniques (a) and (b), and using a layered model in (e) improves the compression ratio to 345:1 at 36.3 [dB]. Technique (f), on the other hand, incurs

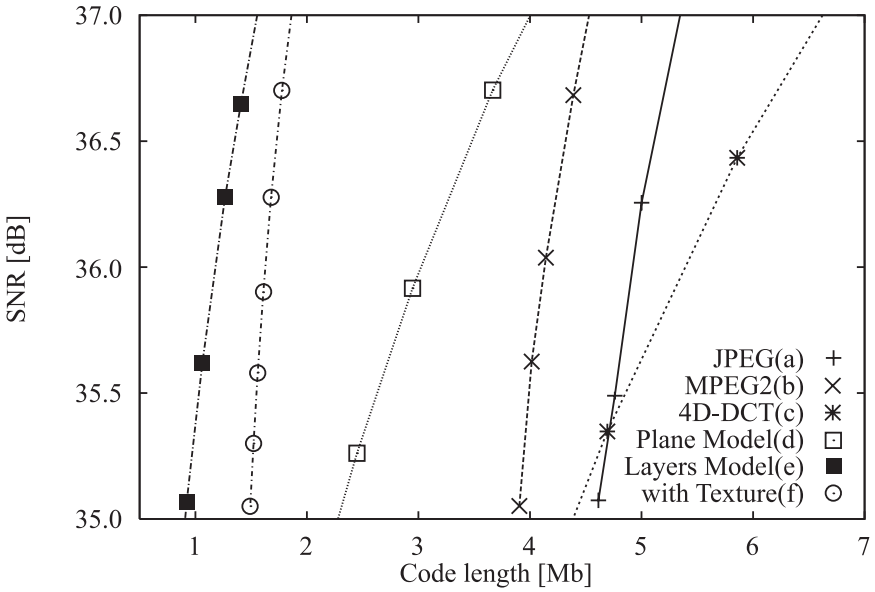


Figure 15. Experimental results .

overhead in texture encoding and a subsequent drop in compression rate compared to (e), but is nevertheless capable of interactive video display. We have therefore confirmed that using redundancy in 4D data space makes for efficient compression surpassing that performed on individual 2D pictures.

6. SAMPLING RAY DATA BY A TELECENTRIC LENS

As described in section 2, 4D data space $f'(P,Q,\theta,\phi)$ can be treated as a set of orthogonal projection images $f'_a(P,Q)$. Accordingly, the use of a telecentric lens that can directly capture orthogonal projection images makes it possible to obtain a 4D data space directly.[19] While sections 3, 4, and 5 have focused on techniques based on the planar recording method, this section describes an experiment using the spherical recording method that can handle orthogonal projection images without distortion.

6.1. Capturing orthogonal projection images

In the experiment, orthogonal projection images of a set of objects are first captured from multiple directions. Some of these images are shown in Fig. 16. Specifically, these are images of two cubes of the same size arranged one behind the other. It can be seen that both objects in a pair are displayed with the same size regardless of the distance from the lens. This confirms that the telecentric lens is taking orthogonal projection images. These images can now be used to configure 4D data space by spherical recording.

6.2. Synthesis of perspective views

Figure 17 shows the results of synthesizing perspective views of the objects described above at various viewpoints by reading out appropriate ray data from 4D data space.

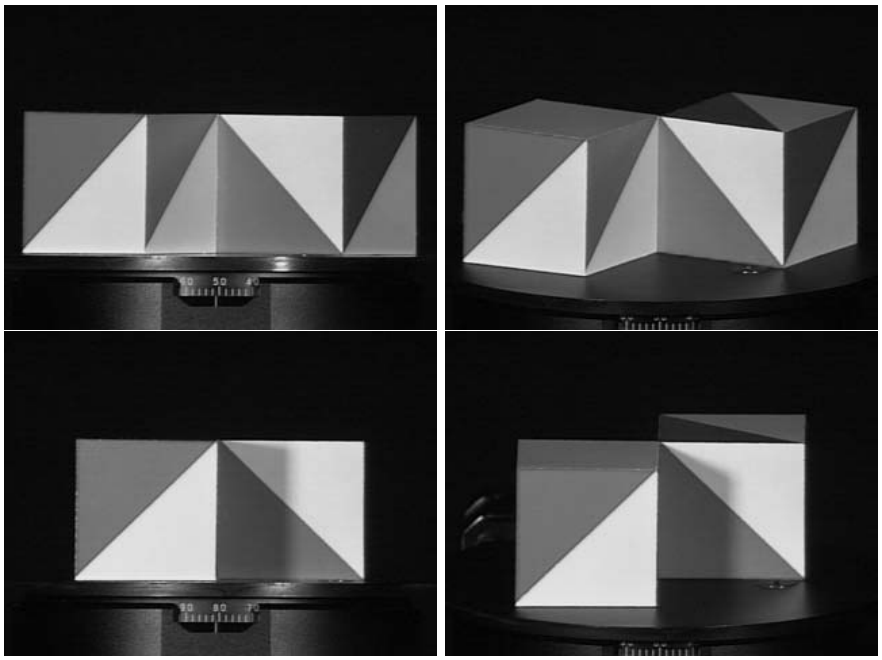


Figure 16. Examples of orthogonal projection images.

We can see here how two objects in a pair are now displayed with different sizes in accordance with their relative positions.

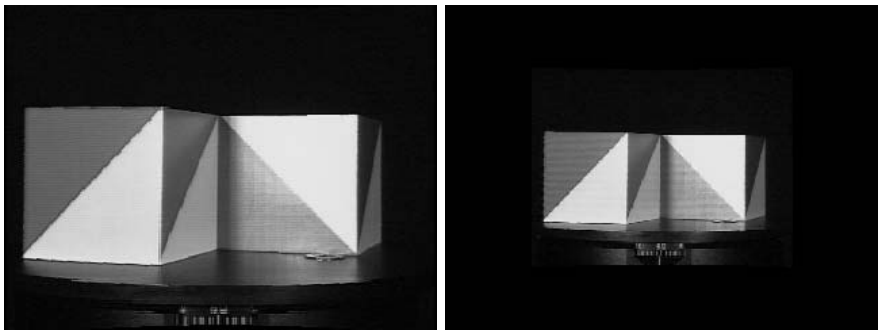


Figure 17. Results of synthesizing perspective projection images from a set of orthogonal projection images.

Capturing a large volume of orthogonal projection images enables perspective views to be synthesized at a variety of viewpoints.

7. CONCLUSION

This paper has presented a technique of representing 3D spatial information based on ray data. Three methods of recording ray data were formalized here: planar

recording, cylindrical recording, and spherical recording. In comparison with past techniques that use perspective projection images, this technique uses orthogonal projection images that enable spatial representation independent of viewpoint. The effectiveness of the proposed technique has been demonstrated experimentally by synthesizing 3D computer graphics using, for example, a real-time video-based rendering system and integral photography. This paper has also proposed the concept of a virtual object surface and has demonstrated its effectiveness in compressing ray data. Finally, the use of a telecentric lens was examined as a way of directly obtaining ray data, and its effectiveness was also demonstrated by experiment.

This ray representation method is a simple and straightforward approach to the problem of how to represent 3D space, an issue shared by various fields including 3D image communications, computer graphics, and virtual reality. In future research, we plan to continue our efforts in systematizing basic theory and to further develop various elemental technologies associated with input systems (camera array, integral photography, and telecentric lenses), compression and transmission methods (virtual object surface, 4D DCT), and interactive display systems (image-based rendering).

ACKNOWLEDGEMENTS

The authors would like to extend their appreciation to NHK Science and Technical Research Laboratories for Fig. 9, an IP image taken by them, and to Tsukuba University for use of their multi-view image database to synthesize Fig. 14.

REFERENCES

1. H. Harashima, "Three-dimensional image coding, future prospect," *Picture Coding Symposium of Japan (PCSJ'92)*, pp. 9 – 12, 1992 (in Japanese).
2. T. Motoki, H. Isono and I. Yuyama, "Present status of three-dimensional television research," Proc. IEEE, **83**, 7, pp. 1009–1021, 1995.
3. T. Naemura, M. Kaneko, and H. Harashima, "Compression and Representation of 3-D Images," IEICE Trans. Inf. & Syst.(Invited Survey Paper), **E82-D**, 3, pp. 558–567, 1999.
4. T. Fujii and H. Harashima, "Data compression and interpolation of multi-view image set", Inst. Electron. Inform. Commun. Eng. (IEICE) Trans. Inf. & Syst., **E77-D**, 9, pp. 987 – 995, 1994.
5. D. V. Papadimitriou and T. J. Dennis, "Stereo in model-based image coding," *Picture Coding Symposium (PCS'94)*, pp. 296–299 (1994).
6. J. R. Ohm and M. E. Izquierdo, "An object-based system for stereoscopic viewpoint synthesis," IEEE Trans. Circuits & Syst. Video Technol., **7**, 5, pp. 801 – 811, 1997.
7. T. Naemura, M. Kaneko, and H. Harashima, "3-D object based coding of multiview images," *Picture Coding Symposium (PCS'96)*, pp. 459 – 464, 1996.
8. D. Tzovaras, N. Grammalidis and M. G. Strintzis, "Object-based coding of stereo image sequences using joint 3-D motion/disparity compression,", IEEE Trans. Circuits & Syst. Video Technol., **7**, 2, pp. 312 – 327, 1997.
9. T. Fujii, "A basic study on the integrated 3-D visual communication", Ph.D thesis, Course of Elec. Eng., The Univ. of Tokyo, 1994 (in Japanese).

10. T. Yanagisawa, T. Naemura, M. Kaneko, and H. Harashima, "Handling of 3-dimensional objects in ray space," *the Inform. and Syst. Society Conf. of Inst. Electron. Inform. Commun. Eng. (IEICE)*, D-169, 1995 (in Japanese).
11. A. Katayama, K. Tanaka, T. Oshino, and H.Tamura, "A viewpoint independent stereoscopic display using interpolating of multi-viewpoint images," *SPIE Stereoscopic displays and virtualreality systems II*, **2409**, pp. 11 – 20, 1995.
12. L. McMillan and G. Bishop, "Plenoptic modeling : An image-based rendering system", *SIGGRAPH'95*, pp. 39 – 46, 1995.
13. M. Levoy and P. Hanrahan, "Light Field Rendering," *SIGGRAPH'96*, pp. 31 – 42, 1996.
14. S. Gortler, R. Grzeszczuk, R. Szeliski and, M. Cohen, "The Lumigraph," *SIGGRAPH'96*, pp. 43 – 54, 1996.
15. T. Naemura, M. Kaneko, and H. Harashima, "Orthographic approach to representing 3-D images and interpolating light rays for 3-D image communication and virtual environment", Signal Process. : Image Commun., **14**, pp. 21 – 37, 1998.
16. T. Naemura and H. Harashima, "Real-Time Video-Based Rendering for Augmented Spatial Communication," *SPIE Visual Commun. and Image Process. (VCIP'99)*, **3653**, pp. 620 – 631, 1999.
17. T. Yoshida, T. Naemura and H. Harashima, "3-D Computer Graphics Based on Integral Photography," *3D Image Conf.*, pp. 35–38, 2000 (in Japanese).
18. T. Takano, T. Naemura, H. Harashima, "3-D space coding using Virtual Object Surface," Inst. Electron. Inform. Commun. Eng. (IEICE) Trans., **J82-D-II**, pp. 1804–1815, 1999 (in Japanese).
19. K.Takeuchi, T.Naemura and H.Harashima, "Acquisition of light rays using telecentric lens," J. Inst. of Image Inform. and Television Eng., **54**, 10, 2000 (to appear in Japanese).
20. E. Adelson and J. Bergen, "The plenoptic function and the elements of early vision", *Computer Models of Visual Processing*, M. Landy and J. Movshon, ed., Chapter 1, MIT Press, 1991.
21. S. E. Chen, "QuickTimeVR - an image-based approach to virtual environment navigation -", *SIGGRAPH'95*, pp. 29 – 38, 1995.
22. T. Fujii, T. Kimoto, and M. Tanimoto, "Data compression of 3-D spatial information based on ray-space coding," J. Inst. of Image Inform. and Television Eng., **52**, 3, pp. 356 – 363, 1998 (in Japanese).
23. R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis : an approach to determining structure from motion," Int'l J. Computer Vision, **1**, pp. 7 – 55, 1987.
24. F. Okano, H. Hoshino, J. Arai and I. Yuyama, "Real-time Pickup Method for a Three-dimensional Image based on Integral Photography," Applied Optics, **36**, 7, pp.1598 – 1603, 1997.
25. H. Hoshino, F. Okano, H. Isono and I. Yuyama, "Analysis of resolution limitation of integral photography," Optical Society of America A, **15**, 8, pp. 2058 – 2065, 1998.

26. T. Takano, T. Naemura, M. Kaneko, and H. Harashima, "3-D space coding based on light ray data - Local expansion of compressed light ray data -," J. Inst. of Image Inform. and Television Eng., **52**, 9, pp. 1321 – 1327, 1998 (in Japanese).