Interactive Four-Dimensional Space Visualization Using Five-Dimensional Homogeneous Processing for Intuitive Understanding

(直感的な理解のための5次元同次処理を用いた4次元空間提示)

Yukihito Sakai (student member)[†], Shuji Hashimoto (member)^{††}

Abstract We have developed an interactive visualization system that enables people to both experience and intuitively understand 4-D space. We proposed a framework in which people can visualize 4-D space using 5-D homogeneous processing. This way of processing uses 5×5 matrices to express various types of transformations and simplifies geometric operations without using any division operations. Therefore, the proposed method improved the system's reliability. We also constructed an interactive environment that associates the user's 3-D position with a 4-D eye point that lies on a 4-D spherical surface surrounding a 4-D solid. The system is composed of a glasses-free 3-D display, a controller pad, and a personal computer. In this environment, we were able to smoothly observe a 4-D solid from an arbitrary 4-D eye point and direction.

Key words: Four-dimensional space, computer graphics, virtual reality, data visualization, cognitive science, mathematical art.

1. Introduction

Our culture and science have been developed by representing various shapes. In particular, 3-D solids have been of interest in many fields of science or art¹⁾. If people can understand higher dimensions, their ways of thinking can be changed, and this will inspire innovative ideas that lead to scientific and cultural advancements.

In order to extend the studies of 3-D solids, we are developing a visualization system for the intuitive understanding of 4-D space and data. Our system deals with 4-D Euclidean space where four axes are orthogonal to each other. The fourth axis is equivalent to the other three axes, and is not limited to the time axis. The final purpose of our work is to provide an interactive environment using computer graphics and virtual reality so that people can experience and understand 4-D space intuitively.

Higher-dimensional solids have been of interest in various fields for many years, not only in the scientific field but also in literature and art. The geometric elements

(vertex, edge, surface, and cell) of various 4-D solids were clarified in the area of geometry²⁾³⁾. A 4-D solid is also called a polytope. A cell corresponds to a 3-D solid. Some scientific works on 4-D geometric operations and the method of drawing a 4-D solid in 2-D or 3-D space have been reported so far^{4)~6)}. In a literary work⁷⁾ written by Edwin A. Abbott, a character named Square who lived in 2-D space was taught the existence of 3-D space by a visitor named Sphere. What is visible to Square is a Circle whose size gradually changes. In an artwork drawn by Salvador Dali, Christ appears in front of a 3-D cross. The surprising fact is that the cross is drawn as an expanded 4-D cube⁸⁾. According to the expanded 4-D solid, we can understand that a 4-D solid is constructed by 3-D solids⁹⁾¹⁰⁾. In architecture, various architectural design motifs are readily obtained from a higher-dimensional solid¹¹⁾¹²⁾. These works suggest the applicability to an architectural modeling and a structural system. In computer graphics, the procedure of folding a 3-D solid in 4-D space will give us a good understanding of 4-D space¹³⁾¹⁴⁾. In the field of virtual reality, 4-D virtual environments are proposed, in which a door suddenly disappears in virtual 3-D space, and a sphere whose size changes appears¹⁵⁾¹⁶⁾.

Although various studies have been performed on 4-D data visualization and recognition so far, it is not easy for people to understand a 4-D object in an intuitive

Received Mar. 2, 2006; Final received Aug. 18, 2006; Accepted Sept. 6, 2006

[†]Major in Pure and Applied Physics, Graduate School of Science and Engineering, Waseda University
(3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan)

^{††}Department of Applied Physics, School of Science and Engineering, Waseda University
(3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan)

manner, because we do not have any experience of or information on 4-D space. For example, when a 4-D cube (hypercube) is displayed in 3-D space from the single direction of the four axes, we can see a 3-D cube which contains another small 3-D cube inside. The 4-D rotation of the 4-D cube displayed in 3-D space is very different from the 3-D rotation of the 3-D cube^{17)–19)}. We can draw various 2-D perspective drawings of a 3-D solid from different 3-D eye points based on our experience in 3-D space. Conversely, we can imagine a 3-D solid from various 2-D perspective drawings. By analogy with this, if we can observe various 3-D perspective drawings of a 4-D solid, we will obtain its 4-D information.

We have constructed an interactive visualization system in order to obtain 4-D space perception by observing a 3-D screen on which a 4-D object is projected from various positions, directions, and distances in 4-D space in real-time. By using such a 4-D display system, the user will be able to move to the desired 4-D position, direction, and distance in a natural and intuitive fashion to imagine the characteristics of various 3-D perspective drawings of a 4-D object.

We make the following demands on the system for intuitive understanding of 4-D space and data.

- Representation of 4-D solids and data: We define a data structure for general representation of arbitrary 4-D solids and data. This will enable us to visualize any 4-D solid and data such as the complex function, the 3-D scalar function, and the 3-D time-series data.
- Observation in 4-D space: The user can observe various 3-D perspective drawings of a 4-D object, while moving in 4-D space freely and looking at the 4-D object from an arbitrary 4-D eye point and direction. This will enable the user to discover the geometric structure of the 4-D solid, the singular point of the complex function, the extremal value of the 3-D scalar function, and the dynamic characteristics of the 3-D object motion.
- Construction of a system: Investigating an intuitive interface with the emphasis on portability and operability, we have constructed an interactive environment that focuses on strolling in 4-D space via human actions in 3-D space. This will be helpful for the user to understand 4-D objects and space intuitively.

In recent years, we have been developing a 4-D visualization system and an interactive environment for intu-

itive understanding of 4-D space and data^{20)–24)}. Moreover, we have introduced a general representation of a 4-D solid with geometric and topological information²⁵⁾. The developed system consists of a glasses-free 3-D display, a controller pad, and a personal computer.

Although some other attempts have been reported for 4-D object visualization and interactive 4-D display, our system is different from these in the generality of 4-D data processing and in the user interface to allow real-time 4-D space traveling as described in the following related sections.

In this paper, we propose a novel display process of 4-D space using 5-D homogeneous processing in order to improve the handling of 4-D solids and data. 5-D homogeneous processing can express various types of transformations using 5×5 matrices, and can be applied to all points including points at infinity to simplify geometric operations without division operation. Therefore, the proposed method improves the reliability of the system in comparison with 4-D Euclidean processing. We construct the 4-D interactive environment by making efficient use of the proposed visualization algorithm. The system can smoothly change the eye point and visual axis in 4-D space, and helpfully handle a 4-D solid to understand 4-D space. Therefore, the user can intuitively obtain a 4-D feeling different from 3-D space perception. Firstly, we define the data structure of a 4-D solid in Section 2. The geometric operation of 4-D space is shown in Section 3. In Section 4, we propose a novel 4-D visualization algorithm using the 4-D eye point and visual axis, and observe 4-D solids. In Section 5, we discuss the movement of the 4-D eye point bounded on the 4-D spherical surface, and examine the interactivity of the developed system. Finally, conclusions are given in Section 6.

2. Data Structure of a 4-D Solid

A 4-D solid is constructed of 3-D solids called cells in 4-D space. The geometric elements of a 4-D solid are vertex, edge, surface, and cell. Each surface is shared by two different cells. Therefore, we lose the distinction of the front and back sides regarding a surface. However, we have the distinction of the front and back sides regarding each cell in 4-D space.

2.1 Description of a 4-D Solid with Geometric and Topological Information

The 4-D geometric information is expressed as $V(k, d)$ in our system. $V(k, d)$ represents the coordinate value of the vertex of a 4-D solid, where k is the global vertex

number of a 4-D solid and d ($= 1, 2, 3, 4$) is the component number of the coordinates.

The 4-D topological information is expressed as $E(l, k_e)$ in the 4-D wire-frame model. $E(l, k_e)$ represents the global vertex number of a 4-D solid, where l is the edge number of a 4-D solid and k_e ($= 1, 2$) is the local vertex number of the end-points of the edge l . This topology associates the edges of a 4-D solid with the global vertex number k of a 4-D solid.

In computing a 4-D solid model, two types of 4-D topological information are necessary. The first topological information is defined as $P(i, k_c)$ regarding cells in a 4-D solid. $P(i, k_c)$ represents the global vertex number k of a 4-D solid, where i is the cell number of a 4-D solid, and k_c is the local vertex number in the cell i . This topology associates the local vertex number of each cell with the global vertex number k of a 4-D solid. The second topological information is represented by $C(i, j, k_s)$ regarding surfaces in each cell of a 4-D solid. $C(i, j, k_s)$ indicates the local vertex number k_c of the cell i , where i is the cell number of a 4-D solid, j is the surface number in the cell i , and k_s is the local vertex number in the surface j . The shape of each cell of a 4-D solid is expressed as the 3-D solid model by this connection topology.

2.2 Memory Design for Representation of Arbitrary 4-D Solids and Data

The geometric and topological information of a 4-D solid requires a huge amount of memory domain. Compared with 3-D solids, the geometric and topological information of 4-D solids are relatively large. Consequently, the static memory allocation is not appropriate memory design.

We implemented a dynamic memory allocation of the geometric and topological information of a 4-D solid. The system can preserve any 4-D solid in the memory domain independent of the sizes of the geometric and topological requirement of the 4-D solid. In such a case, the dynamic memory allocation is suitable for the stable visualization system of 4-D data. The dynamic memory allocation for the geometric and topological information of a 4-D solid allows us to visualize any 4-D solid and data such as the complex function, the 3-D scalar function, and the 3-D time-series data.

3. Geometric Operation of a 4-D Solid Using the 5×5 Matrix

4-D affine transformation is a linear transformation including translation, which is described as the 5×5

matrix. This transformation is given by:

$$\begin{aligned} V^* &= [x^* \ y^* \ z^* \ w^* \ 1] \\ &= [x \ y \ z \ w \ 1]T_A, \end{aligned} \quad (1)$$

where

$$T_A = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & 0 \\ m_{21} & m_{22} & m_{23} & m_{24} & 0 \\ m_{31} & m_{32} & m_{33} & m_{34} & 0 \\ m_{41} & m_{42} & m_{43} & m_{44} & 0 \\ m_{51} & m_{52} & m_{53} & m_{54} & 1 \end{bmatrix}.$$

We call T_A the 4-D affine transformation matrix. For Eq. (1), we introduce 4-D geometric operation matrices of rotation, scaling, and translation in 4-D space in the following.

3.1 Rotation in 4-D Space

The 4-D rotation matrices are represented as follows:

$$\begin{aligned} T_{xy}(\theta) &= T_{xy}(\sin \theta, \cos \theta) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta & 0 \\ 0 & 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (2)$$

$$\begin{aligned} T_{yz}(\theta) &= T_{yz}(\sin \theta, \cos \theta) \\ &= \begin{bmatrix} \cos \theta & 0 & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sin \theta & 0 & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (3)$$

$$\begin{aligned} T_{xz}(\theta) &= T_{xz}(\sin \theta, \cos \theta) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (4)$$

$$\begin{aligned} T_{xw}(\theta) &= T_{xw}(\sin \theta, \cos \theta) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 & 0 \\ 0 & -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (5)$$

$$\begin{aligned} \mathbf{T}_{yw}(\theta) &= \mathbf{T}_{yw}(\sin \theta, \cos \theta) \\ &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (6)$$

$$\begin{aligned} \mathbf{T}_{zw}(\theta) &= \mathbf{T}_{zw}(\sin \theta, \cos \theta) \\ &= \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (7)$$

where θ represents the rotation around xy -, yz -, xz -, xw -, yw -, and zw -planes, respectively.

3.2 Scaling in 4-D Space

The 4-D scaling matrix is expressed as:

$$\mathbf{T}_s(s_x, s_y, s_z, s_w) = \begin{bmatrix} s_x & 0 & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 & 0 \\ 0 & 0 & s_z & 0 & 0 \\ 0 & 0 & 0 & s_w & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

where s_x , s_y , s_z , and s_w represent scalings in the directions of x -, y -, z -, and w -axes, respectively.

3.3 Translation in 4-D Space

The 4-D translation matrix is expressed as:

$$\mathbf{T}_t(t_x, t_y, t_z, t_w) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & t_w & 1 \end{bmatrix}, \quad (9)$$

where t_x , t_y , t_z , and t_w represent translations in the directions of x -, y -, z -, and w -axes, respectively.

4. Visualization Algorithm of 4-D Space Using 5-D Homogeneous Processing

We aim to display 4-D space that allows us to observe a 4-D solid from an arbitrary 4-D eye point and visual axis. Conventional visualization algorithms are divided into two categories. The first one performs perspective projection for a 4-D solid transformed by 4-D geometric operations such as rotation, translation, and scaling^{(26)~(29)}. The second one visualizes various 3-D perspective drawings of a 4-D solid from different eye points in 4-D space⁽³⁰⁾⁽³¹⁾. In the former case, the eye point is fixed. In the latter case, the visual axis is fixed

or the eye point change is limited. Therefore, these visualization algorithms cannot be generalized and systematized to construct a 4-D display system.

Our visualization algorithm performs the 5-D homogeneous processing described by 5-D linear transformation using 5×5 matrices to generalize 4-D graphics⁽³²⁾⁽³³⁾. The 5-D homogeneous processing is effective to represent various transformations in 4-D space such as 4-D projection transformation as Eq. (10).

$$\begin{aligned} V^* &= [X^* \ Y^* \ Z^* \ W^* \ v^*] \\ &= [X \ Y \ Z \ W \ v] \mathbf{T}_P, \end{aligned} \quad (10)$$

where

$$\mathbf{T}_P = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} \end{bmatrix}.$$

The processing can simplify the geometric operations as it does not suffer from the problems of zero division and points at infinity. We call $\mathbf{T}_P$ the 4-D projection transformation matrix.

The proposed algorithm can represent various 3-D perspective drawings of a 4-D solid from an arbitrary 4-D eye point and visual axis. Therefore, we can observe the inside of the 4-D solid and the vanishing points of the 3-D perspective drawing, and can change the 4-D viewing field. Compared with the conventional algorithms, the proposed algorithm is more flexible for use in an interactive 4-D display system. Hereafter, we will mainly use the matrices of Eqs. (2)–(9) shown in Section 3 to represent various important 4-D transformations.

4.1 Change of Eye Point and Visual Axis in 4-D Space

In the 4-D graphics, we propose a visualization algorithm using the eye point (the look-from point) and the look-at point in 4-D space. This algorithm is composed of the view field transformation, the perspective transformation, the display transformation, and the clipping operation in 4-D space.

(1) View Field Transformation from a 4-D World-Coordinate System to a 4-D Eye-Coordinate System

To create a 3-D perspective drawing for a 4-D solid visualization, we set the 4-D eye point $p_f (x_{p_f}, y_{p_f}, z_{p_f}, w_{p_f})$ and the 4-D look-at point $p_a (x_{p_a}, y_{p_a}, z_{p_a}, w_{p_a})$ in the 4-D world-coordinate system, $x_w y_w z_w w_w$. Then, we define the 4-D eye-

Fig. 1 Relation between the 4-D world- and the 4-D eye-coordinate systems.

coordinate system, $x_e y_e z_e w_e$, the origin of which lies at the eye point, and where the w_e -axis is in the direction from the look-at point to the eye point. Additionally, the z_e -axis is parallel to the $z_w w_w$ -plane (see **Fig. 1**).

We firstly translate the origin of the 4-D world-coordinate system to the 4-D eye point. Then, we rotate the translated coordinate system to let the w_1 -axis correspond to the w_e -axis. The rotated coordinate system coincides with the desired 4-D eye-coordinate system. A coordinate system to describe a 4-D solid is transformed from the 4-D world-coordinate system into the 4-D eye-coordinate system. This view field transformation is represented as follows:

$$V_e = [x_e \ y_e \ z_e \ w_e \ 1] \\ = [x_w \ y_w \ z_w \ w_w \ 1] T_v(p_f, p_a), \quad (11)$$

where

$$T_v(p_f, p_a) \\ = T_t(-x_{p_f}, -y_{p_f}, -z_{p_f}, -w_{p_f}) T_{xy}(\sin \alpha, \cos \alpha) \\ T_{yz}(\sin \beta, \cos \beta) T_{xz}(\sin \gamma, \cos \gamma),$$

$$\left. \begin{aligned} \cos \alpha &= \frac{w_{p_f} - w_{p_a}}{\sqrt{(z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \sin \alpha &= \frac{z_{p_f} - z_{p_a}}{\sqrt{(z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \cos \beta &= \frac{\sqrt{(z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}}{\sqrt{(x_{p_f} - x_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \sin \beta &= \frac{x_{p_a} - x_{p_f}}{\sqrt{(x_{p_f} - x_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \cos \gamma &= \frac{\sqrt{(x_{p_f} - x_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}}{\sqrt{(x_{p_f} - x_{p_a})^2 + (y_{p_f} - y_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \sin \gamma &= \frac{y_{p_a} - y_{p_f}}{\sqrt{(x_{p_f} - x_{p_a})^2 + (y_{p_f} - y_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \end{aligned} \right\}.$$

We call T_v the 4-D view field transformation matrix.

(2) Perspective Transformation from a 4-D Eye-Coordinate System to a 3-D Screen-Coordinate System

Let us define the 3-D screen-coordinate system, $x_s y_s z_s w_s$, which locates at a distance h in the direction from the 4-D eye point p_f to the 4-D look-at point

Fig. 2 Relation between the 4-D eye- and the 3-D screen-coordinate systems.

p_a . The 3-D screen is the $x_s y_s z_s$ -space which has a dimension of $2k \times 2k \times 2k$ (see **Fig. 2**).

A 4-D solid is displayed on the 3-D screen by the perspective projection. This operation is called the perspective transformation. The displayed solid is called the 3-D perspective drawing. The 3-D solid appearing on the 3-D screen is what a person in 4-D space would see from their 4-D viewing position. The 4-D projection transformation can be simplified as a 5-D linear transformation as follows:

$$V_p = [X_p \ Y_p \ Z_p \ W_p \ v_p] \\ = [x_e \ y_e \ z_e \ w_e \ 1] T_p(h, k), \quad (12)$$

where

$$T_p(h, k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -k/h \\ 0 & 0 & 0 & k & 0 \end{bmatrix}.$$

The vertex $V_p (X_p, Y_p, Z_p, W_p, v_p)$ represents the coordinates of a 4-D solid after the perspective transformation. The transformed coordinates (x_p, y_p, z_p, w_p) are obtained as follows:

$$x_p = \frac{X_p}{v_p}, \ y_p = \frac{Y_p}{v_p}, \ z_p = \frac{Z_p}{v_p}, \ w_p = \frac{W_p}{v_p}. \quad (13)$$

Each coordinate value of Eq. (13) is normalized in the range of $-1 \leq x_p, y_p, z_p \leq +1$, $0 \leq w_p \leq +1$, respectively. We call T_p the 4-D perspective transformation matrix.

(3) Display Transformation from a 4-D World-Coordinate System to a 3-D Screen-Coordinate System

By combining Eq. (11) with Eq. (12), we transform the data in the 4-D world-coordinate system to the data in the 3-D screen-coordinate system as follows:

$$\begin{aligned} V_s &= [X_s \ Y_s \ Z_s \ W_s \ v_s] \\ &= [x_w \ y_w \ z_w \ w_w \ 1]T_d, \end{aligned} \quad (14)$$

where

$$T_d = T_v(p_f, p_a)T_p(h, k).$$

The 3-D screen coordinates (x_s, y_s, z_s, w_s) are obtained as follows:

$$x_s = \frac{X_s}{v_s}, \ y_s = \frac{Y_s}{v_s}, \ z_s = \frac{Z_s}{v_s}, \ w_s = \frac{W_s}{v_s}, \quad (15)$$

where

$$-1 \leq x_s, y_s, z_s \leq +1, \ 0 \leq w_s \leq +1.$$

Therefore, from Eq. (15), we can see various 3-D perspective drawings (x_s, y_s, z_s) appearing on the 3-D screen from an arbitrary 4-D eye point and direction in 4-D space. We call T_d the 4-D display transformation matrix. Moreover, the 5-D homogeneous processing can be applied to all points including points at infinity. The points at infinity in x_w -, y_w -, z_w -, and w_w -directions are represented as $V_{x_w} (1, 0, 0, 0, 0)$, $V_{y_w} (0, 1, 0, 0, 0)$, $V_{z_w} (0, 0, 1, 0, 0)$, and $V_{w_w} (0, 0, 0, 1, 0)$, respectively. Substituting these points into Eq. (14), we can see the projection of the points at infinity called the principal vanishing point on the 3-D screen. The 3-D perspective drawings are categorized by the number of principal vanishing points; one-, two-, tree-, and four-point 3-D perspective drawings.

(4) Clipping Operation by a 4-D Viewing Pyramid with Eight Boundary Hyperplanes

James F. Blinn and Martin E. Newell applied the 4-D homogeneous coordinates to the clipping of a line segment in 3-D wire-frame models⁽³³⁾⁽³⁴⁾. In the case of the 3-D solid model, it is not sufficient to clip the edges of the 3-D solid. It is necessary to display the inside of the surfaces of a 3-D solid. Ivan E. Sutherland and Gary W. Hodgman proposed the reentrant polygon clipping^{(35)~(37)}.

In the 4-D graphics, we also require the clipping operation to perform Eq. (15) with the 4-D viewing pyramid as shown in **Fig. 3**. The 4-D viewing pyramid is formed by the 4-D eye point and the 3-D screen. The part of the 4-D solid protruding from the viewing pyramid is removed and not displayed on the 3-D screen. The framework of the clipping algorithm generally follows the clipping of a line segment and the reentrant polygon clipping.

Firstly, Eq. (15) is expressed in the 5-D homogeneous coordinates as Eq. (16).

Fig. 3 4-D viewing pyramid formed by the 4-D eye point and the 3-D screen.

$$-v_s \leq X_s, Y_s, Z_s \leq +v_s, \ 0 \leq W_s \leq +v_s. \quad (16)$$

The 4-D viewing pyramid is enclosed with eight boundary hyperplanes. The vertex $V_0 (X_0, Y_0, Z_0, W_0, v_0)$ of a 4-D solid is identified on the inside or the outside of the 4-D viewing pyramid by using the distance coordinates which represent the distances from the vertex to the eight boundary hyperplanes. The distance coordinates are given by:

$$(D_{T_0}, D_{B_0}, D_{R_0}, D_{L_0}, D_{Y_0}, D_{H_0}, D_{N_0}, D_{F_0}), \quad (17)$$

where

$$\left. \begin{aligned} D_{T_0} &= v_0 - Y_0 \\ D_{B_0} &= v_0 + Y_0 \\ D_{R_0} &= v_0 - X_0 \\ D_{L_0} &= v_0 + X_0 \\ D_{Y_0} &= v_0 - Z_0 \\ D_{H_0} &= v_0 + Z_0 \\ D_{N_0} &= v_0 - W_0 \\ D_{F_0} &= W_0 \end{aligned} \right\}.$$

Regarding the two end-points of an edge, if all the distance components of the end-points are not negative, the edge lies in the visible region. However, if there exists at least one pair of negative corresponding distance components of the end-points, the edge lies outside of the visible region. If there exists at least one pair of corresponding distance components with opposite signs, while the other distance components are not negative, a part of the edge is inside the visible region, and the other parts are outside of the visible region. In this case, the edge intersects with the boundary hyperplane. For example, the intersection point: V of an edge $E_0 (V_0 (X_0, Y_0, Z_0, W_0, v_0), V_1 (X_1, Y_1, Z_1, W_1, v_1))$ with each boundary hyperplane is given by:

$$\left. \begin{array}{ll} v_s - Y_s = 0, & V = \overline{D_{T_0} V_1 - D_{T_1} V_0} \\ v_s + Y_s = 0, & V = \overline{D_{B_0} V_1 - D_{B_1} V_0} \\ v_s - X_s = 0, & V = \overline{D_{R_0} V_1 - D_{R_1} V_0} \\ v_s + X_s = 0, & V = \overline{D_{L_0} V_1 - D_{L_1} V_0} \\ v_s - Z_s = 0, & V = \overline{D_{Y_0} V_1 - D_{Y_1} V_0} \\ v_s + Z_s = 0, & V = \overline{D_{H_0} V_1 - D_{H_1} V_0} \\ v_s - W_s = 0, & V = \overline{D_{N_0} V_1 - D_{N_1} V_0} \\ W_s = 0, & V = \overline{D_{F_0} V_1 - D_{F_1} V_0} \end{array} \right\}. \quad (18)$$

By the above-mentioned clipping operation using Eqs. (17) and (18), a 4-D wire-frame model is perfectly represented, while a 4-D solid model representation requires that the surfaces of the cells are closed. In the case of 4-D solid representation, as well as the clipping operation of edges, the end-points of the successive edges of a surface are identified inside or outside of a single boundary hyperplane with the distance coordinates. If some edges of the surface are clipped by the boundary, the vertices and the edges regarding the surface are changed by calculating the intersections to the boundary, and a partially clipped surface is formed. By repeating this process for each boundary hyperplane, the clipping operation of surfaces constructs a surface which fully lies in the visible region.

The number of vertices edges, surfaces, and cells is increased and decreased by the 4-D clipping operation. We need to construct a renewed data structure removing the redundant geometric elements for the appropriate representation of the 4-D solid. By using the distance coordinates, the 4-D clipping algorithm examines the geometric elements of the 4-D solid; the geometric element must be generated, retained, removed, or divided. Due to such a distinction, the new geometric elements are added to the output list of the data structure. Therefore, the unnecessary geometric element does not exist and the memory domain is optimal. The proposed system can dynamically allocate memory to reproduce the data structure $V(k, d)$, $E(l, k_e)$, $P(i, k_c)$, and $C(i, j, k_s)$, efficiently.

4.2 Observation of 4-D Solids from an Arbitrary Eye Point and Visual Axis in 4-D Space

In the following, we show examples of 4-D solids and data from an arbitrary 4-D eye point and visual axis. There are five regular polyhedrons in 3-D space; the tetrahedron, the hexahedron, the octahedron, the dodecahedron, and the icosahedron. On the other hand, there are six regular polytopes (4-D solids) in 4-D space; the 5-cell, the hypercube (the 8-cell), the 16-cell, the

24-cell, the 120-cell, and the 600-cell. In the field of geometry²⁾³⁾, the geometric elements (vertex, edge, surface, and cell) of these 4-D solids are widely-known. The 5-cell, the hypercube, the 16-cell, the 24-cell, the 120-cell, and the 600-cell are constructed by five tetrahedrons, eight hexahedrons, 16 tetrahedrons, 24 octahedrons, 200 dodecahedrons, and 600 tetrahedrons, respectively. Each surface of the 4-D solid is shared by two different cells (3-D solids). In our system, the coordinates of the vertices of these 4-D solids are normalized, as the vertices of these regular polytopes are contained on the 4-D spherical surface. Here, the parameters (h, k) regarding the 3-D screen for perspective projection are given as $(0.5, 0.5)$, respectively.

Fig. 4 shows 4-D solids from different 4-D eye points. As we can see various 2-D perspective drawings of a 3-D solid from different eye points in 3-D space, various 3-D perspective drawings of a 4-D solid appear on the 3-D screen seen from different 4-D eye points.

Fig. 5 shows 4-D solids with the changes of the 4-D visual axis (the direction from the 4-D eye point to the 4-D look-at point). The visual axis change modifies the origin of the 3-D screen and the shape of the 4-D solid visualization. Therefore, we can observe various 3-D perspective drawings by changing the location on the 3-D screen. Our system can display only 4-D solids inside the 3-D screen by the 4-D clipping operation.

When the 4-D eye point approaches a 4-D solid, we can see the inside of the 4-D solid clipped by the 3-D screen as shown in **Fig. 6**. By this representation, we will understand that the 4-D solid is constructed by cells, and that each surface is shared by two different cells.

Fig. 7 shows the projection of the points at infinity. We take the hypercube as an example of a 4-D solid. By using 5-D homogeneous processing, the points at infinity in x_w -, y_w -, z_w -, and w_w -directions are all displayed on the 3-D screen. Different colors are given to the four vanishing points VP_{x_w} , VP_{y_w} , VP_{z_w} , and VP_{w_w} , respectively. We will intuitively be conscious of the vanishing points as the 4-D axes.

Visualization examples of the complex functions $f(z)$ are shown in **Fig. 8**. The input complex variables $z = x + iy$ correspond to (x, y) in the range of $-\pi \leq x, y \leq +\pi$, and the output complex variables $f(z) = u(x, y) + iv(x, y)$ correspond to (z, w) . Using our system, the singular point can be found easily, which suggests that our system is applicable for mathematical education.

Fig. 4 4-D solids from different 4-D eye points.

Fig. 5 4-D solids with the change of the 4-D visual axis.

5. Movement Algorithm for Construction of 4-D Interactive Environment

There are a few reports concerning interaction with 4-D solids. In the existing interactive environments, keyboard input or a joystick associated with human actions are often used for geometric operations of a 4-

Fig. 6 The inside of 4-D solids clipped by the 3-D screen.

Fig. 7 Projections of the points at infinity. (Principal vanishing points)

Fig. 8 Visualization examples of complex functions.

D solid⁽¹⁵⁾⁽¹⁶⁾⁽³⁸⁾, which enable a user to experience the rotation of the 4-D solid in 3-D space. Some papers have introduced a haptic interface for data exploration in higher-dimensional space to recognize the distribution of higher-dimensional data⁽³⁹⁾⁽⁴⁰⁾. However, in these methods, the user's 4-D viewing position is fixed. Therefore, the user is not allowed to move around the

4-D object in 4-D space. We investigate a novel interactive method for intuitive interaction with 4-D space, and aim to induce 4-D space perception by allowing the user to travel in 4-D space.

We can observe 4-D solids from an arbitrary 4-D eye point and visual axis by the proposed visualization algorithm. However, we have some difficulties in moving around in 4-D space to interact with 4-D solids, since the action in 3-D space is not directly related to the movement in 4-D space. In order to overcome such difficulties, we propose a novel 4-D movement algorithm which makes the human actions in 3-D space correspond to the movement and the rotation in 4-D space bounded on a 4-D spherical surface, and have developed a real-time and interactive environment using intuitive interfaces such as a glasses-free 3-D display and a controller pad like a flight simulator.

The 4-D movement algorithm is implemented by extending the 4-D visualization algorithm shown in Section 4. The framework of the proposed 4-D interactive method is consistently constructed by 5-D homogeneous processing.

Using the proposed system, we can smoothly observe any 4-D solid and data from various 4-D positions, directions, and distances extending the experiences of 3-D space. Therefore, we can acquire the unusual sense and phenomenon of 4-D space, which is different from 3-D space. The user will be able to feel the depth in the direction of the fourth axis from the various 3-D perspective drawings changed (pushed, pulled, made to appear, or made to vanish) in 3-D space.

5.1 Basic Concept for Movement in 4-D Space

We make 3-D space correspond to a 4-D spherical surface. Before considering the 4-D case, let us begin with a study of the 3-D case. Consider the correspondence between 2-D space and a 3-D spherical surface. As shown in **Figs. 9, 10**, when a 2-D person moves by the distance d_3 in the upper (head) direction after moving along the circumference of the radius r around the origin by the distance d_2 ($= r\theta$) from the point $(r, 0)$, the corresponding 3-D person moves by distance d_3 ($= r\phi$) in the upper (head) direction along the 3-D spherical surface after the rotation in the direction of θ ($= d_2/r$) at the point $(0, 0, r)$. For example, when the 2-D person moves by the distance $2\pi r$ in the upper direction from the point $(r, 0)$ in 2-D space, the 3-D person comes back to the starting point $(0, 0, r)$ on the 3-D spherical surface. By this mapping, we can make 2-D space correspond to

Fig. 9 Movement of the 2-D person in 2-D space.

Fig. 10 Rotation and movement of the 3-D person on the 3-D spherical surface.

all points on the 3-D spherical surface. According to the movements in 2-D space, a 2-D perspective drawing projected from a 3-D solid changes periodically.

We extend this 3-D analogy to the 4-D case. That is, we consider the correspondence between 3-D space and a 4-D spherical surface. We define 3-D space where the user exists and assume a 4-D sphere of radius r surrounding a 4-D solid. In the same way as the 3-D case, we associate the point $(0, 0, 0, r)$ which lies on the 4-D spherical surface with the point $(0, 0, r)$ of 3-D space. When the 3-D person moves by the distance d_4 in the upper (head) direction after moving by the distance (d_2, d_3) ($= (r\theta, r\phi)$) in the direction of (θ, ϕ) along the 3-D spherical surface from the point $(0, 0, r)$, the 4-D person moves by the distance d_4 ($= r\psi$) in the upper

(head) direction along the 4-D spherical surface after the rotation in the direction of $(\theta, \phi) = (d_2/r, d_3/r)$ at the point $(0, 0, 0, r)$. For example, when the 3-D person moves by the distance $2\pi r$ in the upper direction from the point $(0, 0, r)$ in 3-D space, the 4-D person comes back to the starting point $(0, 0, 0, r)$ on the 4-D spherical surface.

As we perform the movement and the rotation on the 4-D spherical surface making use of the experiences of 3-D space, it is expected that the proposed algorithm will be useful for the induction of a perception of 4-D space.

5.2 Correspondence between 3-D Space and a 4-D Sphere

We will further explain the correspondence between 3-D space and a 4-D sphere. In order to make the movement on the 3-D screen correspond to the movement in 4-D space bounded on a 4-D spherical surface, we introduce the 3-D eye point on the 3-D screen corresponding to the 4-D eye point, and the respective upper directions for the observing persons in the 3-D and 4-D viewing pyramid. Here, we assume a 3-D person who observes a 4-D solid projected onto the 3-D screen with the 3-D eye point. Similarly, we assume a 4-D person who observes a 4-D solid with the 4-D eye point. The 3-D person moves in the horizontal direction and the upper direction. The 4-D person moves in the upper direction. The 3-D eye point is initially located on the z_s -axis of $x_s y_s z_s$ -space. The coordinates are $p_{f-s} (0, 0, r)$. The 4-D eye point is initially located on the w_w -axis of the 4-D world-coordinate system, $x_w y_w z_w w_w$. The coordinates are $p_f (0, 0, 0, r)$. The 3-D and 4-D upper directions are initially in the direction of the y_w -axis and are given as $u_s [0 \ r \ 0]$ and $u_e [0 \ r \ 0 \ 0]$, respectively (see Figs. 11, 12).

The 4-D person bounded on the 4-D spherical surface is allowed to take two types of actions. One is rotation on the 4-D spherical surface and the other is movement in their 4-D upper direction along the 4-D spherical surface. In the proposed movement algorithm, the rotation of the 4-D person on the 4-D spherical surface corresponds to the movement of the 3-D person bounded on the 3-D spherical surface, while the movement of the 4-D person along the 4-D spherical surface corresponds to the movement of the 3-D person in their 3-D upper direction in 3-D space.

Firstly, let us discuss the rotation of the 4-D person. The rotation of the 4-D person on the 4-D spherical surface has two degrees of freedom (θ, ϕ) , since the ro-

Fig. 11 Movement of the 3-D person in 3-D space.

Fig. 12 Rotation and movement of the 4-D person on the 4-D spherical surface.

tation is performed in 3-D space bounded on the 4-D spherical surface. As shown in Fig. 11, when the 3-D person moves the distance $d_2 (= r\theta)$ in their horizontal direction along the 3-D spherical surface from the initial position on the 3-D sphere, the coordinates are transformed as follows:

$$\begin{aligned} p_{f-s} &= [x_{p_{f-s}} \ 0 \ z_{p_{f-s}} \ 0 \ 1] \\ &= [0 \ 0 \ r \ 0 \ 1] T_{yw}(\theta), \end{aligned} \quad (19)$$

where

$$\theta = \frac{d_2}{r}.$$

Next, we assume that the 3-D person moves the distance $d_3 (= r\phi)$ in their upper direction of the 3-D viewing pyramid along the 3-D spherical surface. The transformation is represented as Eq. (20). In similar fashion, we calculate the additional movement Δd_3 . Combining Eq. (20) with Eq. (21), we can obtain the 3-D eye direction vector e_s of the 3-D viewing pyramid in 3-D space as described in Eq. (22). The 3-D eye direction vector e_s is orthogonal to the 3-D upper direction and its value depends on the movement. Coincident with the 3-D person's movement, the 4-D person rotates at the 4-D position of the 4-D spherical surface (see Fig. 12).

$$\begin{aligned} u_s &= [x_{u_s} \ y_{u_s} \ z_{u_s} \ 0 \ 0] \\ &= [0 \ r \ 0 \ 0 \ 0] \mathbf{T}_{xw}(\phi) \mathbf{T}_{yw}(\theta), \end{aligned} \quad (20)$$

where

$$\begin{aligned} \phi &= \frac{d_3}{r}. \\ u_{s-a} &= [x_{u_{s-a}} \ y_{u_{s-a}} \ z_{u_{s-a}} \ 0 \ 0] \\ &= [0 \ r \ 0 \ 0 \ 0] \mathbf{T}_{xw}(\phi) \mathbf{T}_{yw}(\theta), \end{aligned} \quad (21)$$

where

$$\begin{aligned} \phi &= \frac{d_3 + \Delta d_3}{r}. \\ e_s &= u_{s-a} - u_s \\ &= [x_{e_s} \ y_{e_s} \ z_{e_s} \ 0 \ 0]. \end{aligned} \quad (22)$$

Secondly, let us discuss the movement of the 4-D person along the 4-D spherical surface. After the above-mentioned movement bounded on the 3-D spherical surface, the 3-D person moves the distance d_4 in the upper direction of the 3-D viewing pyramid in 3-D space (see Fig. 11), while the 4-D person moves the distance $d_4 (= r\psi)$ in the upper direction of the 4-D viewing pyramid along the 4-D spherical surface (see Fig. 12), correspondingly. The coordinates are transformed as expressed in Eq. (23). In similar fashion, we calculate the additional movement Δd_4 . Combining Eq. (23) with Eq. (24), we can obtain the 4-D up vector u_e of the 4-D viewing pyramid in 4-D space as described in Eq. (25). The direction of the 4-D up vector u_e is the 4-D upper direction and its value depends on the movement.

$$\begin{aligned} p_f &= [x_{p_f} \ y_{p_f} \ z_{p_f} \ w_{p_f} \ 1] \\ &= [0 \ 0 \ 0 \ r \ 1] \mathbf{T}_{xz}(\psi) \mathbf{T}_{xw}(\phi) \mathbf{T}_{yw}(\theta), \end{aligned} \quad (23)$$

where

$$\psi = \frac{d_4}{r}.$$

$$\begin{aligned} p_{f-a} &= [x_{p_{f-a}} \ y_{p_{f-a}} \ z_{p_{f-a}} \ w_{p_{f-a}} \ 1] \\ &= [0 \ 0 \ 0 \ r \ 1] \mathbf{T}_{xz}(\psi) \mathbf{T}_{xw}(\phi) \mathbf{T}_{yw}(\theta), \end{aligned} \quad (24)$$

where

$$\begin{aligned} \psi &= \frac{d_4 + \Delta d_4}{r}. \\ u_e &= p_{f-a} - p_f \\ &= [x_{u_e} \ y_{u_e} \ z_{u_e} \ w_{u_e} \ 0]. \end{aligned} \quad (25)$$

Moreover, we formulate the shifts of the 4-D spherical radius and the changes of the 4-D look-at point. As shown in **Fig. 13**, when the 3-D person moves away from the origin of the 3-D screen, the 4-D person shifts in the radial direction of the 4-D sphere from r to r' (in a position anterior to r). Conversely, when the 3-D person approaches the origin of the 3-D screen, the 4-D person enters into the 4-D solid transparently, moving in the radial direction of the 4-D sphere from r to r' (in a position posterior to r), which allows us to observe the details of inside the 4-D solid.

The origin of the 3-D screen-coordinate system coincides with the 3-D look-at point p_{a-s} . When the 3-D person changes the 3-D visual axis, or the look-at point p_{a-s} ($x_{p_{a-s}}, y_{p_{a-s}}, z_{p_{a-s}}$) in 3-D space, the 4-D visual axis, or the look-at point p_a of the 4-D person on the 4-D spherical surface is changed correspondingly (see **Fig. 14**). When the 3-D look-at point p_{a-s} on the 3-D screen varies in the range of $-r \leq x_{p_{a-s}}, y_{p_{a-s}}, z_{p_{a-s}} \leq +r$, the coordinates of the 4-D look-at point p_a are transformed in the following equation:

$$\begin{aligned} p_a &= [x_{p_a} \ y_{p_a} \ z_{p_a} \ 0 \ 1] \\ &= [0 \ 0 \ 0 \ 0 \ 1] \mathbf{T}_t(x_{p_{a-s}}, y_{p_{a-s}}, z_{p_{a-s}}, 0). \end{aligned} \quad (26)$$

Due to the changes of Eq. (26), we can observe a 4-D solid out of the 4-D viewing pyramid.

5.3 Procedure for Sequential Movement in 4-D Space

In iterating movement to achieve an arbitrary position, direction, and distance in 4-D space, the 4-D eye point p_f , the 4-D look-at point p_a , the 4-D up vector u_e on the 4-D spherical surface, and the 3-D eye direction vector e_s in the 3-D screen need to be transformed to the data of the 4-D world-coordinate system.

Firstly, we allocate the 4-D world coordinates of the 4-D eye point p_{f-b} , the 4-D up vector u_{e-b} , and the 3-D eye direction vector e_{s-b} in advance of the movement as follows:

$$p_{f-b} = (x_{p_{f-b}}, y_{p_{f-b}}, z_{p_{f-b}}, w_{p_{f-b}}). \quad (27)$$

Fig. 13 Relation between the movement of the 3-D person in the radial direction of the 3-D sphere and the movement of the 4-D person in the radial direction of the 4-D sphere.

Fig. 14 Relation between the change of the 3-D visual axis of the 3-D person on the 3-D spherical surface and the change of the 4-D visual axis of the 4-D person on the 4-D spherical surface.

$$u_{e,b} = [x_{u_{e,b}} \ y_{u_{e,b}} \ z_{u_{e,b}} \ w_{u_{e,b}}]. \quad (28)$$

$$e_{s,b} = [x_{e_{s,b}} \ y_{e_{s,b}} \ z_{e_{s,b}} \ w_{e_{s,b}}]. \quad (29)$$

From Eq. (27), the view field transformation matrix is represented as follows:

$$\begin{aligned} & T_v(p_{f,b}, 0) \\ &= T_t(0, 0, 0, 0) T_{xy}(\sin \alpha, \cos \alpha) \\ & \quad T_{yz}(\sin \beta, \cos \beta) T_{xz}(\sin \gamma, \cos \gamma), \end{aligned} \quad (30)$$

where

$$\left. \begin{aligned} \cos \alpha &= \frac{w_{p_{f,b}}}{\sqrt{z_{p_{f,b}}^2 + w_{p_{f,b}}^2}} \\ \sin \alpha &= \frac{z_{p_{f,b}}}{\sqrt{z_{p_{f,b}}^2 + w_{p_{f,b}}^2}} \\ \cos \beta &= \frac{\sqrt{z_{p_{f,b}}^2 + w_{p_{f,b}}^2}}{\sqrt{x_{p_{f,b}}^2 + z_{p_{f,b}}^2 + w_{p_{f,b}}^2}} \\ \sin \beta &= \frac{-x_{p_{f,b}}}{\sqrt{x_{p_{f,b}}^2 + z_{p_{f,b}}^2 + w_{p_{f,b}}^2}} \\ \cos \gamma &= \frac{\sqrt{x_{p_{f,b}}^2 + y_{p_{f,b}}^2 + z_{p_{f,b}}^2 + w_{p_{f,b}}^2}}{\sqrt{x_{p_{f,b}}^2 + y_{p_{f,b}}^2 + z_{p_{f,b}}^2 + w_{p_{f,b}}^2}} \\ \sin \gamma &= \frac{-y_{p_{f,b}}}{\sqrt{x_{p_{f,b}}^2 + y_{p_{f,b}}^2 + z_{p_{f,b}}^2 + w_{p_{f,b}}^2}} \end{aligned} \right\}.$$

Now, we define two coordinate systems, the 4-D viewing-coordinate system and the 3-D viewing-coordinate system regarding the 4-D viewing pyramid and the 3-D viewing pyramid, respectively. The first one takes the y_e -axis as the 4-D upper direction. The second one takes the y_s -axis as the 3-D upper direction.

By multiplying Eq. (28) by Eq. (30), we can transform $u_{e,b}$ to $u_{e,b}^*$ described in the 4-D eye-coordinate system as follows:

$$\begin{aligned} u_{e,b}^* &= [x_{u_{e,b}}^* \ y_{u_{e,b}}^* \ z_{u_{e,b}}^* \ w_{u_{e,b}}^* \ 0] \\ &= [x_{u_{e,b}} \ y_{u_{e,b}} \ z_{u_{e,b}} \ w_{u_{e,b}} \ 0] T_v(p_f, 0). \end{aligned} \quad (31)$$

Then, for the 4-D viewing-coordinate system, from Eq. (31), we obtain the transformation matrix regarding the 4-D up vector $u_{e,b}$ as follows:

$$\begin{aligned} & T_{u_e}(u_{e,b}^*) \\ &= T_{xw}(\sin \eta, \cos \eta) T_{zw}(\sin \omega, \cos \omega), \end{aligned} \quad (32)$$

where

$$\left. \begin{aligned} \cos \eta &= \frac{y_{u_{e,b}}^*}{\sqrt{y_{u_{e,b}}^{*2} + z_{u_{e,b}}^{*2}}} \\ \sin \eta &= \frac{-z_{u_{e,b}}^*}{\sqrt{y_{u_{e,b}}^{*2} + z_{u_{e,b}}^{*2}}} \\ \cos \omega &= \frac{\sqrt{y_{u_{e,b}}^{*2} + z_{u_{e,b}}^{*2}}}{\sqrt{x_{u_{e,b}}^{*2} + y_{u_{e,b}}^{*2} + z_{u_{e,b}}^{*2}}} \\ \sin \omega &= \frac{x_{u_{e,b}}^*}{\sqrt{x_{u_{e,b}}^{*2} + y_{u_{e,b}}^{*2} + z_{u_{e,b}}^{*2}}} \end{aligned} \right\}.$$

We call T_{u_e} the 4-D upper direction transformation matrix.

By multiplying Eq. (29) by Eqs. (30) and (32), we can transform $e_{s,b}$ to $e_{s,b}^*$ described in the 4-D viewing-coordinate system as follows:

$$\begin{aligned} e_{s,b}^* &= [x_{e_{s,b}}^* \ y_{e_{s,b}}^* \ z_{e_{s,b}}^* \ w_{e_{s,b}}^* \ 0] \\ &= [x_{e_{s,b}} \ y_{e_{s,b}} \ z_{e_{s,b}} \ w_{e_{s,b}} \ 0] T_v(p_{f,b}, 0) \\ & \quad T_{u_e}(u_{e,b}^*). \end{aligned} \quad (33)$$

Therefore, for the 3-D viewing-coordinate system, from Eq. (33), we obtain the transformation matrix regarding the 3-D eye direction vector $e_{s,b}$ as follows:

$$\mathbf{T}_{e_s}(e_{s,b}^*) = \mathbf{T}_{yw}(\sin \zeta, \cos \zeta), \quad (34)$$

where

$$\left. \begin{aligned} \cos \zeta &= \frac{z_{e_{s,b}}^*}{\sqrt{x_{e_{s,b}}^{*2} + z_{e_{s,b}}^{*2}}} \\ \sin \zeta &= \frac{-x_{e_{s,b}}^*}{\sqrt{x_{e_{s,b}}^{*2} + z_{e_{s,b}}^{*2}}} \end{aligned} \right\}.$$

We call $\mathbf{T}_{e_s}$ the 3-D upper direction transformation matrix.

From Eqs. (22), (23), (25), and (26), we obtain the desired data represented in the 4-D world-coordinate system with the inverse matrices Eqs. (30), (32), and (34) as follows:

$$\begin{aligned} e_s &= [x_{e_s} \ y_{e_s} \ z_{e_s} \ w_{e_s} \ 0] \\ &= [x_{e_s} \ y_{e_s} \ z_{e_s} \ 0 \ 0] \mathbf{T}_{e_s}(e_{s,b}^*)^{-1} \\ &\quad \mathbf{T}_{u_e}(u_{e,b}^*)^{-1} \mathbf{T}_v(p_{f,b}, 0)^{-1}. \end{aligned} \quad (35)$$

$$\begin{aligned} p_f &= [x_{p_f} \ y_{p_f} \ z_{p_f} \ w_{p_f} \ 1] \\ &= [x_{p_f} \ y_{p_f} \ z_{p_f} \ w_{p_f} \ 1] \mathbf{T}_{e_s}(e_{s,b}^*)^{-1} \\ &\quad \mathbf{T}_{u_e}(u_{e,b}^*)^{-1} \mathbf{T}_v(p_{f,b}, 0)^{-1}. \end{aligned} \quad (36)$$

$$\begin{aligned} u_e &= [x_{u_e} \ y_{u_e} \ z_{u_e} \ w_{u_e} \ 0] \\ &= [x_{u_e} \ y_{u_e} \ z_{u_e} \ w_{u_e} \ 0] \mathbf{T}_{e_s}(e_{s,b}^*)^{-1} \\ &\quad \mathbf{T}_{u_e}(u_{e,b}^*)^{-1} \mathbf{T}_v(p_{f,b}, 0)^{-1}. \end{aligned} \quad (37)$$

Considering the change ratio of the radius Δr , we obtain the following 4-D look-at point:

$$\begin{aligned} p_a &= [x_{p_a} \ y_{p_a} \ z_{p_a} \ w_{p_a} \ 1] \\ &= [x_{p_a} \ y_{p_a} \ z_{p_a} \ 0 \ 1] \mathbf{T}_{e_s}(e_{s,b}^*)^{-1} \mathbf{T}_{u_e}(u_{e,b}^*)^{-1} \\ &\quad \mathbf{T}_v(p_{f,b}, 0)^{-1} \mathbf{T}_s(\Delta r, \Delta r, \Delta r, \Delta r), \end{aligned} \quad (38)$$

where

$$\Delta r = \frac{r'}{r}.$$

Finally, we introduce the 3-D eye direction vector e_s of Eq. (35) and the 4-D up vector u_e of Eq. (37) into the 4-D display transformation of Eq. (14) using Eqs. (36) and (38). The conclusive operation of the 4-D display transformation is represented as follows:

$$\begin{aligned} V_s &= [X_s \ Y_s \ Z_s \ W_s \ v_s] \\ &= [x_w \ y_w \ z_w \ w_w \ 1] \mathbf{T}_d, \end{aligned} \quad (39)$$

where

$$\mathbf{T}_d = \mathbf{T}_v(p_f, p_a) \mathbf{T}_{u_e}(u_e^*) \mathbf{T}_{e_s}(e_s^*) \mathbf{T}_p(h, k),$$

$$\begin{aligned} \mathbf{T}_v(p_f, p_a) &= \mathbf{T}_t(-x_{p_f}, -y_{p_f}, -z_{p_f}, -w_{p_f}) \mathbf{T}_{xy}(\sin \alpha, \cos \alpha) \\ &\quad \mathbf{T}_{yz}(\sin \beta, \cos \beta) \mathbf{T}_{xz}(\sin \gamma, \cos \gamma), \end{aligned}$$

$$\left. \begin{aligned} \cos \alpha &= \frac{w_{p_f} - w_{p_a}}{\sqrt{(z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \sin \alpha &= \frac{z_{p_f} - z_{p_a}}{\sqrt{(z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \cos \beta &= \frac{\sqrt{(z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}}{\sqrt{(x_{p_f} - x_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \sin \beta &= \frac{x_{p_f} - x_{p_a}}{\sqrt{(x_{p_f} - x_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \cos \gamma &= \frac{\sqrt{(x_{p_f} - x_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}}{\sqrt{(x_{p_f} - x_{p_a})^2 + (y_{p_f} - y_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \\ \sin \gamma &= \frac{y_{p_f} - y_{p_a}}{\sqrt{(x_{p_f} - x_{p_a})^2 + (y_{p_f} - y_{p_a})^2 + (z_{p_f} - z_{p_a})^2 + (w_{p_f} - w_{p_a})^2}} \end{aligned} \right\},$$

$$\begin{aligned} u_e^* &= [x_{u_e}^* \ y_{u_e}^* \ z_{u_e}^* \ w_{u_e}^* \ 0] \\ &= [x_{u_e} \ y_{u_e} \ z_{u_e} \ w_{u_e} \ 0] \mathbf{T}_v(p_f, p_a), \end{aligned}$$

$$\mathbf{T}_{u_e}(u_e^*) = \mathbf{T}_{xw}(\sin \eta, \cos \eta) \mathbf{T}_{zw}(\sin \omega, \cos \omega),$$

$$\left. \begin{aligned} \cos \eta &= \frac{y_{u_e}^*}{\sqrt{y_{u_e}^{*2} + z_{u_e}^{*2}}} \\ \sin \eta &= \frac{-z_{u_e}^*}{\sqrt{y_{u_e}^{*2} + z_{u_e}^{*2}}} \\ \cos \omega &= \frac{\sqrt{y_{u_e}^{*2} + z_{u_e}^{*2}}}{\sqrt{x_{u_e}^{*2} + y_{u_e}^{*2} + z_{u_e}^{*2}}} \\ \sin \omega &= \frac{x_{u_e}^*}{\sqrt{x_{u_e}^{*2} + y_{u_e}^{*2} + z_{u_e}^{*2}}} \end{aligned} \right\},$$

$$\begin{aligned} e_s^* &= [x_{e_s}^* \ y_{e_s}^* \ z_{e_s}^* \ w_{e_s}^* \ 0] \\ &= [x_{e_s} \ y_{e_s} \ z_{e_s} \ w_{e_s} \ 0] \mathbf{T}_v(p_f, p_a) \mathbf{T}_{u_e}(u_e^*), \end{aligned}$$

$$\begin{aligned} \mathbf{T}_{e_s}(e_s^*) &= \mathbf{T}_{yw}(\sin \zeta, \cos \zeta), \\ \left. \begin{aligned} \cos \zeta &= \frac{z_{e_s}^*}{\sqrt{x_{e_s}^{*2} + z_{e_s}^{*2}}} \\ \sin \zeta &= \frac{-x_{e_s}^*}{\sqrt{x_{e_s}^{*2} + z_{e_s}^{*2}}} \end{aligned} \right\}. \end{aligned}$$

The 3-D screen coordinates (x_s, y_s, z_s, w_s) are obtained as follows:

$$x_s = \frac{X_s}{v_s}, \ y_s = \frac{Y_s}{v_s}, \ z_s = \frac{Z_s}{v_s}, \ w_s = \frac{W_s}{v_s}, \quad (40)$$

where

$$-1 \leq x_s, y_s, z_s \leq +1, \ 0 \leq w_s \leq +1.$$

Therefore, from Eq. (40), we can smoothly observe various 3-D perspective drawings (x_s, y_s, z_s) of a 4-D solid and data on the 3-D screen. The consistent algorithm using the 5-D homogeneous processing is well suited to construct a 4-D interactive system which makes it easy to understand 4-D space and data. The proposed system will give the user the 4-D space perception from the changes of a 4-D solid displayed in 3-D space through the interaction using the user's 3-D action. For example, different 3-D objects can be seen, when traveling in 4-D space toward a 4-D object.

The proposed algorithm makes the origin of the 4-D

Fig. 15 Setup of the 4-D interactive visualization system.

world-coordinate system correspond to the center of a 4-D sphere. Therefore, the user is allowed to move in 4-D space bounded on the 4-D sphere focusing around the origin of the 4-D world-coordinate system.

5.4 Observation of 4-D Solids in an Interactive Environment

We constructed an intuitive and interactive 4-D space display system which made human actions in 3-D space correspond to the movement of the eye point in 4-D space. The developed system consists of three parts: a main processor, an input interface device, and an output interface device. As a main processor, we adopted the personal computer VAIO Type R (Sony) installed the operating system Windows XP (Microsoft) and the proposed algorithms implemented with the Software Development Kits (SDKs) Visual C++ (Microsoft), the 3-D graphics library OpenGL (Silicon Graphics), and the I/O interface DirectX (Microsoft). As an input interface device, we employed the flight controller FLIGHT SIM YOKE (CH) for intuitive operation. As an output interface device, we used the glasses-free 3-D display X3D-20 Display AD (X3D Technologies) through OpenGL Enhancer (X3D Technologies) for intuitive motion perception. The glasses-free 3-D display shows various images to the different angles; using a parallax, we can see as a 3-D image by accepting a different image with each eye. These interface devices are connected into the main processor (see **Fig. 15**).

In the following, we show the hypercube images as a typical example of a 4-D object. The number of vertices, edges, surfaces, and cells of a hypercube is 16, 32, 24, and 8, respectively²⁾³⁾. The coordinates of the vertices

of the hypercube are represented by the permutation of $(\pm 1, \pm 1, \pm 1, \pm 1)$. In our system, the coordinates of the hypercube are the normalized data. Therefore, the coordinates of the vertices of the hypercube are defined as the permutation of $(\pm 0.5, \pm 0.5, \pm 0.5, \pm 0.5)$. A cell corresponds to a 3-D solid. In the hypercube, the cell represents a 3-D cube. The hypercube is constructed so that four of eight cells are connected at each vertex, three of those are connected at each edge, and two of those are connected at each surface. Different colors are given to each cell of the 3-D solid model in our system to make for easy perception.

As shown in **Fig. 16**, two steering actions of the flight controller are associated with the movement on the 3-D spherical surface, the first one regarding turning action, and the second one regarding forward and backward action. These actions correspond to the movement in the 3-D horizontal direction and the movement in the 3-D upper direction along the 3-D spherical surface as **a** and **b** in **Fig. 16**, respectively. **Fig. 17** shows the view of a projected 4-D solid corresponding to the rotation at a point on the 4-D spherical surface. In this figure, the numbers of the images correspond to the action numbers in **Fig. 16**. Although each hypercube image of the bottom shown in **Fig. 17** looks the same, the 4-D upper direction is changed by the steering actions shown in **Fig. 16**. The 4-D person can understand the 3-D perspective drawing of a 4-D solid at a glance, however, the 3-D person cannot obtain a perfect view of a 3-D solid without moving around on the 3-D spherical surface. We made the 3-D person's motion on the 3-D spherical surface correspond to the 4-D person's rotation on the 4-D spherical surface.

Two buttons on the handle are used for movement control. As shown in **Fig. 18**, the right button on the handle is associated with the movement in the 3-D upper direction in 3-D space. When the 3-D person moves by the distance $2\pi r$ from the starting point, the 4-D person goes around the 4-D spherical surface. **Fig. 19** shows the view corresponding to the motion on the 4-D spherical surface. In this movement, the shape of the 4-D solid projected onto the 3-D screen is changed. We can observe that the hypercube is constructed of eight cells with different colors. The perfect understanding of a 4-D solid can be obtained from various 3-D perspective drawings in the sequential motion of the 4-D eye point. We can perceive the motion bounded on the 4-D spherical surface through the motion in the upper direction of the 3-D viewing pyramid in 3-D space.

As shown in **Fig. 20**, the left button on the handle is associated with the movement in the radial direction of the 3-D sphere. This action corresponds to changes of the 4-D spherical radius. When we intend to obtain a microscopic or macroscopic view of a target, we will move toward or move away from the target. We implemented this motion by changing the radius of the 4-D sphere. We can even enter inside a 4-D solid and observe it from the inside by the 4-D clipping operation of the system (see **Fig. 21**).

Three levers of the flight controller are associated with the 3-D visual axis, or the coordinate values of the look-at point in 3-D space as **a**, **b**, and **c** in **Fig. 22**. **Fig. 23** shows the changes of the look-at point in 4-D space which correspond to the changes of the look-at point in 3-D space. We can move around on the 4-D spherical surface, while changing the 4-D visual axis, and observe the outside of the 4-D viewing field by this process.

Our system can perform the above-mentioned processes simultaneously, which enables us to intuitively move in 4-D space and observe 4-D solids in real-time.

Using the proposed system, the user can smoothly observe the various 3-D perspective drawings of a 4-D solid changed (pushed, pulled, made to appear, or made to vanish) by the movements on the 4-D spherical surface, and will gradually acquire information about the 4-D viewing position and a knowledge of the 4-D space structure from the characteristics of one-, two-, tree-, and four-point 3-D perspective drawings. The system can effectively give the user whole new sensations different from 3-D space perception. Therefore, we will be able to travel intuitively on the desired 4-D viewing position based on the 4-D space perception.

6. Conclusions

We proposed a novel visualization algorithm of 4-D space using 5-D homogeneous processing, and developed an interactive visualization system for the intuitive understanding of 4-D space and data, which provides an interactive environment for people to experience and understand 4-D space. We were able to perform a novel 4-D interactive system development which makes extensive use of 5-D homogeneous processing. In the system, we can smoothly observe a 4-D solid from an arbitrary 4-D eye point and direction, and handle a 4-D solid with intuitive interface devices to understand 4-D space. Further considerations include an evaluation experiment to prove the effectiveness of the proposed system. We expect that our 4-D visualization system

Fig. 16 Movements of the 3-D eye point along the 3-D spherical surface. (Control actions) The numbers of the actions correspond to the image numbers in **Fig. 17**.

Fig. 17 Changes of the 4-D upper direction on the 4-D spherical surface. (Hypercube images) The numbers of the images correspond to the action numbers in **Fig. 16**.

will be effective not only for 4-D solid visualization but also for data visualization, robot motion planning in 4-D space-time, cognitive science, and mathematical art.

Challenges for the future encompass the construction of 4-D solid contents (4-D labyrinths, 4-D building

Fig. 20 Movements of the 3-D eye point in the radial direction of the 3-D sphere. (Control actions) The numbers of the actions correspond to the image numbers in Fig. 21.

Fig. 22 Changes of the 3-D visual axis in 3-D space. (Control actions) The numbers of the actions correspond to the image numbers in Fig. 23.

Fig. 21 Movements of the 4-D eye point in the radial direction of the 4-D sphere. (Hypercube images) The numbers of the images correspond to the action numbers in Fig. 20.

Fig. 23 Changes of the 4-D visual axis in 4-D space. (Hypercube images) The numbers of the images correspond to the action numbers in Fig. 22.

- 2) H.S.M. Coxeter: "Introduction to Geometry Second Edition," ISBN 0-471-50458-0, John Wiley & Sons (February 1989).
- 3) H.S.M. Coxeter: "Regular Polytopes," ISBN 0-486-61480-8, Dover (June 1973).
- 4) Koji Miyazaki, Kenichi N. Ishihara: "Introduction to Four-Dimensional Graphics," *Forma*, **6**, 2, pp. 85-97 (1991).
- 5) Koji Miyazaki: "Visualization of Four-Dimensional Objects," *Forma*, **9**, 3, pp. 225-231 (1994).
- 6) Motonaga Ishii: "On a General Method to Calculate Vertices of N-dimensional Product-Regular Polytopes," *Forma*, **14**, 3, pp. 221-237 (1999).
- 7) Edwin A. Abbott: "Flatland: A Romance of Many Dimensions," ISBN 0-451-52290-7, New American Library (March 1984).
- 8) Salvador Dali: "Corpus Hypercubus (Crucifixion)," Metropolitan Museum of Art, New York (1954).
- 9) Ricardo Perez Aguila: "4D Orthogonal Polytopes," B.S. Dissertation, Universidad de las Americas — Puebla (December 2001).
- 10) Antonio Aguilera Ramirez, Ricardo Perez Aguila: "A Method for Obtaining the Tesseract by Unraveling the 4D Hypercube," *Journal of the 10th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision — WSCG 2002*, **10**, 1, pp. 1-8 (February 2002).
- 11) Hiroaki Nakata: "The Possibility of Architectural Designing Through the Medium of 4 Dimensional Hypersphere," *AIJ Journal of Architecture, Planning and Environmental Engineering*, **478**, pp. 107-112 (December 1995), (in Japanese).
- 12) Hiroya Tanaka, Keiko Sato: "Architectural Digital Design derived from Higher-Dimensional Polytopes," *The 1999 Annual Meeting — The Architectural Institute of Japan*, pp. 549-550 (September 1999), (in Japanese).
- 13) Keimei Kaino: "Folding Tetrahedra and Four-Dimensional Origami," *Forma*, **14**, 1, pp. 49-56 (2000).
- 14) Akira Inoue, Ryouko Itohara, Kuniaki Yajima, Keimei Kaino: "CG Image Generation of Four-Dimensional Origami," *The Journal of the Society for Art and Science*, **4**, 4, pp. 151-158 (December 2005).
- 15) Michael D'Zmura, Philippe Colantoni, Gregory D. Seyranian: "Virtual Environments with Four or More Spatial Dimensions," *Presence*, **9**, 6, pp. 616-631 (December 2000).
- 16) Michael D'Zmura, Philippe Colantoni, Gregory D. Seyranian: "Visualization of Events from Arbitrary Spacetime Perspectives," *Proceedings of the SPIE/SPSE Symposium on Electronic Imaging: Visual Data Exploration and Analysis*, **3960**, pp. 35-40 (January 2000).
- 17) Alexander K. Dewdney: "COMPUTER RECREATIONS: A Program for Rotating Hypercubes Induces Four-Dimensional Dementia," *Scientific American*, pp. 8-13 (April 1986).
- 18) Michio Kaku, Jennifer T. Thompson: "Beyond Einstein: The Cosmic Quest for the Theory of the Universe," ISBN 0-553-34349-1, Bantam Dell Pub Group (March 1987).
- 19) Thomas F. Banchoff: "Beyond the Third Dimension: Geometry, Computer Graphics, and Higher Dimensions (Scientific American Library Series)," ISBN 0-716-75025-2, W. H. Freeman & Company (August 1990).
- 20) Makoto Murata, Shuji Hashimoto: "Interactive Environment for Intuitive Understanding of 4D data," *Proceedings of Advanced Robot and Virtual Reality — ISMCR 2000*, pp. 165-170 (September 2000).
- 21) Makoto Murata, Shuji Hashimoto: "Interactive Environment for Intuitive Understanding of 4D Object and Space," *Proceedings of the International Conference on Multimedia Modeling — MMM 2000*, pp. 383-401 (November 2000).
- 22) Shinichiro Sanji, Makoto Murata, Pitoyo Hartono, Shuji Hashimoto: "A Virtual Reality System for High-Dimensional Data Visualization," *Proceedings of the 2001 IEEE International Conference on Multimedia and Expo — ICME 2001*, Paper ID 527 (August 2001).
- 23) Yukihiro Sakai, Shuji Hashimoto: "Visualization System for Intuitive Understanding of 4-D Object and Space," *Proceedings of the 68th National Convention of IPSJ*, pp. 4-49-4-50 (March 2006), (in Japanese).
- 24) Yukihiro Sakai, Shuji Hashimoto: "VR System for Intuitive Understanding of 4-D Objects and Space," *Proceedings of the 2006 JSME Conference on Robotics and Mechatronics — ROBOMECH 2006*, pp. 1P1-D15(1)-1P1-D15(4) (May 2006), (in Japanese).
- 25) Yukihiro Sakai, Shuji Hashimoto: "4D Solid Representation for Visualization," *Proceedings of the 10th International Conference on VIRTUAL SYSTEMS and MULTIMEDIA — VSMM 2004*, pp. 593-601 (November 2004).
- 26) Sachiko Miyazoe, Reiji Tsuruno, Sadahiko Nagae, Setsuo Fukunaga: "Modeling and its Displaying in 4-Dimensional Space," *IPSJ SIGNotes Computer Graphics and CAD*, **23-6**, pp. 1-9 (October 1986), (in Japanese).
- 27) Barbara Hausmann, Hans-Peter Seidel: "Visualization of Regular Polytopes in Three and Four Dimensions," *Proceedings of EUROGRAPHICS'94*, **13**, 3, pp. C305-C316 (September 1994).
- 28) Andrew J. Hanson: "Computer Graphics beyond the Third Dimension," *Course Notes for SIGGRAPH'98* (July 1998).
- 29) Svatopluk Zacharias, Daniela Velichova: "Projection from 4D to 3D," *Journal for Geometry and Graphics*, **4**, 1, pp. 55-69 (2000).
- 30) Koji Miyazaki, Keiichi Ishihara: "Four-Dimensional Graphics," ISBN 4-254-10075-2, Asakura-Shoten Publishers (September 1989), (in Japanese).
- 31) Steven R. Hollasch: "Four-Space Visualization of 4D Objects," M.S. Dissertation, Arizona State University (August 1991).
- 32) Yukihiro Sakai, Hiroshi Murata, Fujio Yamaguchi: "3-D Solid Reconstruction from Spherical Images: Comparative Study between Euclidean and Four-Dimensional Processing," *Proceedings of the JSPE Annual Meeting 2001 Autumn*, p. 42 (September 2001), (in Japanese).
- 33) Fujio Yamaguchi: "Computer-Aided Geometric Design: A Totally Four-Dimensional Approach," ISBN 4-431-70340-3, Springer-Verlag Tokyo (November 2002).
- 34) James F. Blinn, Martin E. Newell: "Clipping Using Homogeneous Coordinates," *Proceedings of SIGGRAPH'78*, pp. 245-251 (August 1978).
- 35) Ivan E. Sutherland, Gary W. Hodgman: "Reentrant Polygon Clipping," *Communications of the ACM*, **17**, 1, pp. 32-42 (January 1974).
- 36) David F. Roger: "Procedural Elements for Computer Graphics," ISBN 0-070-53548-5, William C Brown Pub (October 1997).
- 37) James D. Foley, Andries Van Dam, Steven K. Feiner, John F. Hughes: "Computer Graphics: Principles and Practice (Systems Programming Series)," ISBN 0-201-84840-6, Addison-Wesley Pub (Sd) (August 1995).
- 38) Julieta C. Aguilera: "Virtual Reality and the Unfolding of Higher Dimensions," *Proceedings of SPIE*, **6055**, Stereoscopic Displays and Virtual Reality Systems XIII, pp. 605-612 (January 2006).
- 39) Steven Feiner, Clifford Beshers: "Worlds within Worlds: Metaphors for Exploring n-Dimensional Virtual Worlds," *Proceedings of the ACM SIGGRAPH Symposium on User Interface Software and Technology (UIST'90)*, pp. 76-83 (October 1990).
- 40) Wataru Hashimoto, Hiroo Iwata: "Multi-Dimensional Data Browser with Haptic Sensation," *Transactions of the Virtual Reality Society of Japan*, **2**, 3, pp. 9-16 (September 1997), (in Japanese).

Yukihiro Sakai received his B.Eng. and M.Eng. degrees in Mechanical Engineering from Waseda University, Tokyo, Japan, in 1999 and 2001, respectively. From 2001 to 2004, he worked as a system engineer for Toshiba Corporation. Since 2004, He has been a Ph.D. student in the Graduate School of Science and Engineering, Waseda University. His research interests include cognitive science, mathematical art, and computer graphics.

Shuji Hashimoto received his B.S., M.S. and Dr.Eng. degrees in Applied Physics from Waseda University, Tokyo, Japan, in 1970, 1973 and 1977, respectively. He is currently a Professor in the Department of Applied Physics, School of Science and Engineering, Waseda University. Since 2000, he has been a director at the Humanoid Robotics Institute, Waseda University. From 1979 to 1991, he was with the Faculty of Science, Toho University, Funabashi. He was the Vice President of the International Computer Music Association until 2001. His research interests are in human communication and Kansei information processing including image processing, music systems, neural computing, and humanoid robotics. Since 2003, he has served as the chair of the chapter promotion committee, IEEE Tokyo Section.