

INTERACTIVE ENVIRONMENT FOR INTUITIVE UNDERSTANDING OF 4D OBJECT AND SPACE

Makoto MURATA Shuji HASHIMOTO

Department of Applied Physics, School of Science and Engineering,
WASEDA University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555
JAPAN

{ makoto, shuji }@shalab.phys.waseda.ac.jp
<http://www.phys.waseda.ac.jp/shalab/>

We present a new virtual reality system to assist intuitive understanding of 4D data. The system is composed of a motion capture device, a large screen, a stereo viewing system and a graphics workstation. In this environment, a user can go around the 4D world freely and interact with 4D data.

The system can display four types of 4D data including 4D solid objects, 3D scalar potential fields, the profile of the complex function and 3D time-series data. Two mapping methods are employed to generate corresponding 3D images: They are the projection from 4D space to 3D space and the 3D section of 4D data.

The user can observe the 4D object from an arbitrary 4D viewpoint and slice it from an arbitrary 4D direction by matching his 3D position to the point which lies on the surface of the 4D sphere surrounding the object. Moreover, using stereo glasses and position sensors attached to each hand, the user can easily rotate the 3D shape mapped from 4D space.

As a result of evaluation, it was proved that the interaction with 4D data is effective for intuitive understanding of 4D.

Keywords : four dimensions, virtual reality, computer graphics

1 INTRODUCTION

High dimensional data are often treated in various research fields, for example statistics, multimedia database, image processing, mathematics and so on. If we can visually understand the nature of the data or intuitively find the tendency, it will be easy to design the system. However, it is hard for us to grasp high dimensional data, so let us consider the problem from low dimension.

We can imagine the 3D shape from a 2D image of an object. Actually, many studies which focus on the recovery of 3D shape information from a 2D image[1] have been done in the field of computer vision. In addition, the recent 3D computer graphics technology, using many rendering algorithm, can represent not only shape information but also lighting, color, material and reflective information, so that we can reconstruct 3D information from a 2D image.

As the relationship between 3D space and 2D space can be extended mathematically to the higher dimensional case, it may be expected to make 4D information visible using 3D graphics. However, we can not easily understand 4D space because we are living in the 3D world.

The difference between the 4D case and the 3D case is very clear. In the 3D case, we already have a knowledge of the 3D world, so we can easily understand the 3D shape

from a 2D image. On the other hand, we do not have any experience in the 4D world because we have never been in the 4D world. Therefore, it should be said that the estimation of the 4D shape from a 3D image is not a recognitive problem but a cognitive problem.

For intuitive understanding of 4D space, we need to train ourselves to acquire a knowledge and experience of 4D space. If we can understand 4D space intuitively, we will also understand 4D data easily. Furthermore, we may understand higher dimensional data in the same way. By the development of recent computer graphics and virtual reality technologies, it has become possible to build such a training system.

In this paper, we propose a novel system, as shown in *figure 1*, to let everyone understand 4D data. In addition, we introduce interactive techniques to make the user's 3D position correspond to the surface of the 4D sphere.

This paper is organized as follows. We discuss related works and describe our approach and goals in Section 2. We will give an overview of our system in section 3. Algorithm for visualizing 4D data is introduced in Section 4. Our interface using motion capture is discussed in Section 5. In Section 6, we show examples of four type of 4D data. Results of evaluation are shown in section 7. Finally, we conclude the paper with a summary and proposals for future research in Section 8.

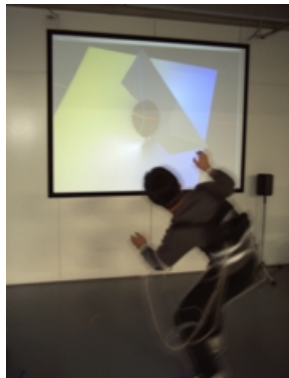


Figure 1. A user in Interactive Environment

2 APPROACH

2.1 Related Works

A study on four dimensions was begun in the field of mathematics. For example, in the field of Topology, 4D objects are treated as 3D manifolds and investigated with properties based on their mappings, singular point and symmetries and so on. These studies established the logical basics of 4D research[2].

From the 70's, using 3D computer graphics, studies on displaying 4D data were begun. Thomas Banchoff developed the 3D projection and section techniques to display the 4D geometry by using computer graphics[3]. He applied these basic techniques to hyper cubes, mathematical surfaces, functions, curves and so on[4]. He proposed the use of color mapped from the value of the fourth coordinate. However,

the graphics hardware did not have enough power for the interactive animation at that time.

In the 80's, David Banks studied the manipulation of 2D surfaces in 4D space[5] and on the basic problem of illumination in 4D space[6].

In the 90's, A. J. Hanson proposed several 4D rendering algorithms which were carefully extended from the 3D algorithm. He summarized the theory of 4D geometry and rotation[7,8]. He also investigated the 4D light source and visualized quite a realistic 4D cube and many mathematical objects which reflect 4D shading information by extending 3D algorithms such as the Gouraud and Phong shading[9]. Toriwaki visualized the 4D reflection, using the 4D extension of the ray-tracing and volume rendering [10].

These works have enabled us to observe realistic images of 4D data represented in 3D. However, if an algorithm which requires a large amount of computational power is used, the complexity of computation will be larger in 4D than 3D.

Recently, Geomview and Meshview which visualize mathematical 4D data were proposed[11,12]. By extending studies on user interface of manipulating a 3D object[13,14] to the 4D case, Geomview and Meshview enabled us to observe 4D mathematical data interactively. Using auxiliary information which is represented on sub-windows, Geomview can display 3D projections of 4D geometry. Meshview is a compact and portable viewer to display 3D projections of 4D mathematical functions and hyper-cubes.

2.2 Research Goals

As mentioned in the previous section, the existing approaches visualizing 4D data are categorized into two types. The first approach is to create a realistic 3D image of 4D projection by extending the 3D rendering algorithm. The second approach is to focus on the interaction between a user and 4D data.

We employ the second approach. We put emphasis on the interaction between a user and a 4D object rather than to create a realistic 4D image. We consider the real-time interaction to be more effective to understand 4D space than the static observation of a realistic still image.

Compared with the related works, our approach is new in the display method and the interface. In the display method, we do not use only the 3D projection of 4D data but also the 3D slice of 4D data. Incorporating two methods, the user can select the appropriate method to display 4D data. In manipulation interface, a motion capture is employed as the input interface so that the user can more intuitively manipulate 4D objects. In addition, a large screen and stereo glasses are employed as the output interface so that the user can more directly feel the 4D world.

We obtain 4D information only from the 3D shape. If the method of visualization changes, the same 4D data must be represented into the different style. Therefore, it is also necessary to investigate relations between the type of 4D data and its presentation method.

Our research goal is to construct an interactive environment which enables us to understand the 4D world intuitively. Even if a user could not understand the 4D world at first, it is expected that the user will be able to understand through the interaction with 4D objects in our interactive environment.

3 System Overview

An overview of our system is shown in *Figure 2*. A user who puts on position sensors walks around the front of the large screen. According to the position of user and movement of both hands, a 3D image mapped from 4D is drawn on the screen. We use stereo glasses to perceive the 3D shape easily. It is expected that the user intuitively interacts with 4D data as if he were in the 4D world through these systems.

The flow of process is shown in *Figure 3*. We use a motion capture device as an input interface, which is a wireless and 6 degree-of-freedom measurement system for the multiple points on user's body based on magnetic fields. This device works in real-time and transmits data via Ethernet. We use position data of three sensors which are attached to user's head and both hands. The captured position data are transmitted to SGI Onyx2 (MIPS 250MHz x 2) via 100baseT network. In Onyx2, the position data of user's head are mapped to the 4D view point or the direction of slice according to the display method. The method for mapping 3D position data into 4D will be discussed in Section 5. Using this information, our system draws the projection or the section of 4D data in real-time. The details of the display method will be explained in the next section. We use OpenGL to draw these images on the 100 inch single screen.

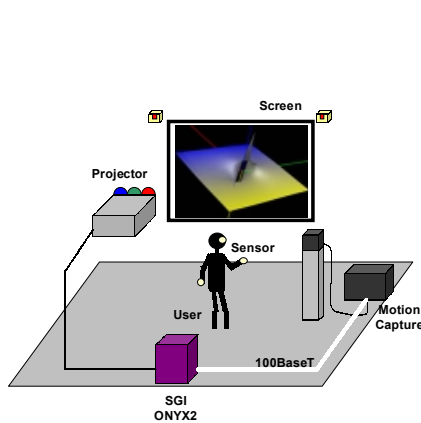


Figure 2. System Overview

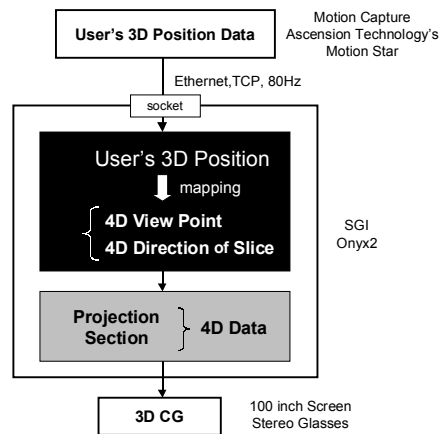


Figure 3. Flow diagram of our system

4 DISPLAY METHODS

In this section, we will discuss two methods to display 4D data in 3D space. The first is the projection method. The second is the slice method. These methods are the natural extension of methods used to display 3D data in 2D space. Our system deals with the 4D Euclidean space where four axes are perpendicular to each other. The fourth axis is equivalent to the other three axes and is not limited to the time axis.

4.1 The Projection Method

The projection method is the 4D extension of the general 3D rendering algorithm. The projection is the natural and powerful method to display 3D objects in a 2D plane. A 3D analogy of the projection method is shown in *Figure 4*. When the position of the eye and screen are defined, a 3D object can be projected onto the 2D screen. In the 4D case, the screen becomes a 3D one. According to the change of user's position, the 4D object will show a variety of 3D shape. If we approach the 4D object, we will see the 3D shape larger. Moreover, we may enter the inside of the 4D object. It is expected that these interactive changes of the 3D shape becomes effective for intuitive understanding of a 4D object.

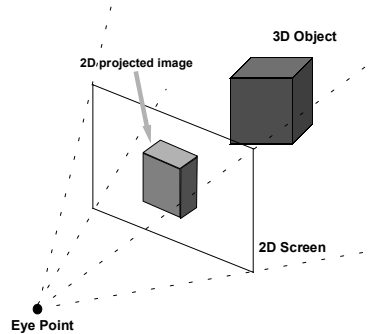


Figure 4. 3D Analogy of the Projection Method

As we focus on the interaction between the user and 4D data, we do not consider the 4D light source as this would increase the complexity of computation. We are interested in real-time interactive operations such as the change of an observational point in 4D space.

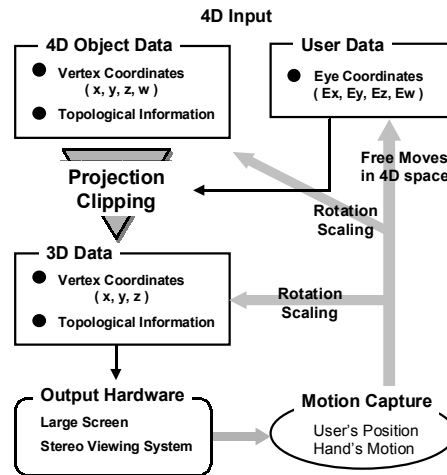


Figure 5. Flow Chart of the Projection Method

The flow chart of this method is depicted in *Figure 5*. Data input to the system are 4D object data and the viewpoint coordinates of the user. The 4D object data include whole vertex coordinates and the topological link information about each polygon. Using the viewpoint coordinates, we project the 4D vertex coordinates onto 3D space. Equations of the projection are shown in equation (1) and (2). Topological information about each polygon is conserved. Therefore, we can draw the 3D shape generated from 4D data as a binocular stereo image on the screen. To increase the reality of 3D shape, we use a large (100 inch) screen and stereo glasses. Moreover, the user can rotate and enlarge the projected 3D object in 3D space.

Projective Transformation

$$\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} -Ey/A & Ex/A & 0 & 0 \\ -ExEz/AB & -EyEz/AB & A/B & 0 \\ -ExEw/B & -EyEw/B & -EzEw/B & B \\ Ex & Ey & Ez & Ew \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} \quad (1)$$

$$A = \sqrt{Ex^2 + Ey^2} \quad B = \sqrt{1 - Ew^2}$$

$$x' = \frac{t}{t+w}x \quad y' = \frac{t}{t+w}y \quad z' = \frac{t}{t+w}z \quad (2)$$

4D coordinates of vertices are represented as (X,Y,Z,W). (x', y', z') and (Ex, Ey, Ez, Ew) represents the 3D coordinates of vertices and the 4D unit vector which is directed to the viewpoint, respectively. t represents the distance between the origin and a viewpoint.

4D Simple Clipping

Using the transformation described above, we can arbitrarily move the user's position in the 4D world in real-time. Thus, we will enter even the inside of 4D objects transparently. However, if we do not clip the object which is behind the eye point, the object, which can not be seen actually, would be seen. The general 3D CG algorithm employs the clipping algorithm which uses six (top, bottom, right, left, near, far) clip planes to avoid such a problem. In the 4D case, 6 clip planes become 8 clip solids because the 4D cube has 8 solids.

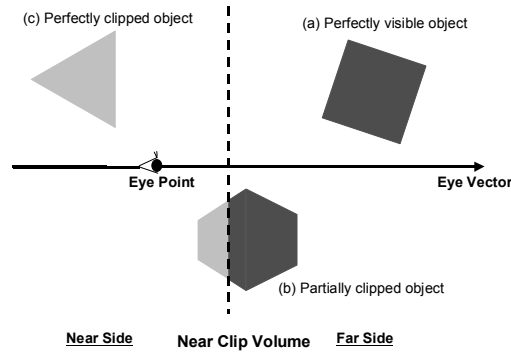


Figure 6. 4D Simple Clipping

For simplification of algorithm and reduction of computational complexity, we introduce a 4D simple clipping algorithm which uses only one plane for clipping. This algorithm uses only near clip volume. If we use only a near clip volume, we can judge whether each vertex is before the near clip or not, by only using the “w”-coordinate calculated in the equation (1).

As shown in *Figure 6*, clipping is classified into the following three cases determined by the relation between the object and the near clip.

- case (a) : If the object exists on the far side of the near clip volume, it is perfectly drawn.
- case (b) : If the object exists across the near clip volume, part of the object is clipped, by calculating the point of intersection between object's edge and the volume. Using these intersections, we draw only a visible part of polygons.
- case (c) : If the object exists on the near side of the near clip volume, it is perfectly clipped, and never drawn.

After the projective transformation described in the equation (1), each polygon is processed by the clipping algorithm described above. Finally, all visible 4D vertices are perspectively converted to vertices by the equation (2).

Interface and Controls

We use a motion capture device as an input interface. The captured data are mapped onto the main controls of the projection. We can also use the mouse and keyboard as an input interface. They corresponds to other optional controls. All control of the projection method is summarized to *Table 1*. The detail of the motion capture interface will be described in Section 5.

Controls	Purpose	User's Motion or Operation	Interface
Move in 4D	To observe the object from the 4D arbitrary point	Walking around the 3D space	Motion Capture
Rotation in 3D	To understand the 3D shape generated from the 4D object	Altering the direction to connect both hands	Motion Capture
Rotation in 4D	To observe the object from the 4D arbitrary direction	Rotating the 4D object around xy, yz, zw, xz, xw, yw planes	GUI
Translation in 4D	To understand the direction of each 4D axis	Translating the 4D object in the direction of X, Y, Z, W axes	GUI
Scaling in 3D	To observe the 3D shape with appropriate scale	Dragging the mouse	GUI
Drawing Mode	To observe the object with appropriate drawing mode	Selecting polygon or wireframe mode	GUI

Table 1. All Controls of the Projection Method

4.2 The Slice Method

The slice method is to slice the 4D object by a 3D volume which we call a hyper plane in 4D space. We show a 3D analogy of the slice method in Figure 7. The section of a 3D object and a 2D plane is a polygon. According to the translation of the 2D plane, the 2D shape of the section changes. In the 4D case, the section of a 4D object and a hyper plane is a 3D solid object. According to the translation of the hyper plane in 4D space, the 3D shape of the section shows various transformations. It is expected that the animation of the moving section will give us an insight into understanding 4D objects.

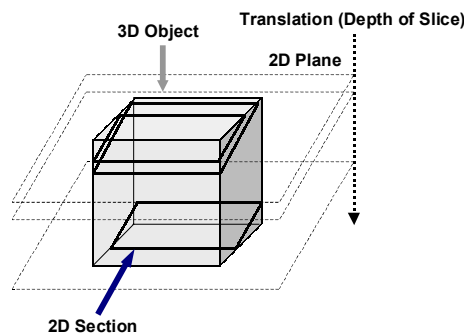


Figure 7. 3D Analogy of the slice method

It is not necessary to consider the 4D light source because the section does not reflect 4D light information. We emphasize interactive operations including the translation of the hyper plane, the 3D rotation of the section and the 3D enlargement of the section. The flow chart of the slice method is shown in Figure 8.

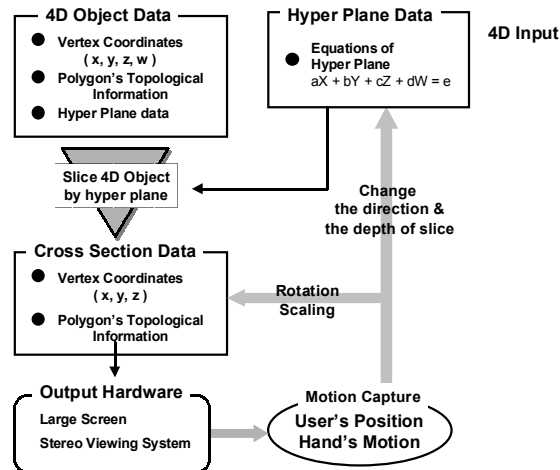


Figure 8. Flow Chart of the Slice Method

Data input are both 4D object data and hyper plane data. 4D object data includes whole vertex coordinates, topological link information about each polygon and information about solids which compose the 4D object. Using these input data, we calculate the section of a 4D object and a hyper plane. Output data is shape data of the section which includes 3D coordinates of all vertices and link information about each 3D polygon. Because the section is a 3D solid object, we draw a 3D section as a binocular stereo image on the screen to understand its shape intuitively.

Interface and Controls

To understand 4D shape information intuitively, various interactive controls are also available in this method. For example, the user can change the direction of slice freely. When the direction of slice is changed, the 3D shape also changes. The direction of slice is defined by a normal vector of a hyper plane. The user can also alter the depth of slice. When the user alters the depth of slice along with a normal vector of a hyper plane, the section shows various kinds of animation. It is expected that these animation images become effective for the user to understand 4D data. To observe the section from an arbitrary direction in 3D, the user can rotate it interactively. These interactive controls are supported by the motion capture interface explained in the next section. All controls of the slice method are summarized in *Table 2*.

Controls	Purpose	User Motion	Interface
Direction of Slice in 4D	To slice the object from an arbitrary direction	Walking around 3D space	Motion Capture
Depth of Slice in 4D	To observe an animation of the section	Changing the distance between both hands	Motion Capture
Rotation in 3D	To understand the 3D shape of the section	Changing the direction to connect both hands	Motion Capture
Scaling in 3D	To observe the 3D shape with appropriate scale	Optional keyboard operation	GUI
Drawing Mode	To observe the object with appropriate drawing mode	Selecting polygon or wireframe drawing mode	GUI

Table 2. All Controls of the Slice Method

5 MOTION CAPTURE INTERFACE

5.1 Correspondence between controls and motion capture

In this chapter, we will discuss our input interface using a motion capture. We use 3 sensors attached to user's head and both hands to measure position data of the user. The sensor attached to user's head is used to represent a 3D user position in 4D space. Sensors attached to both hands are mainly used to manipulate the 3D projection or section in 3D space. All correspondences are summarized in *Table 3*.

In case of the projection method, the sensor attached to the head corresponds to the 4D viewpoint which lies on the surface of a 4D sphere. By defining the 4D sphere surrounding the target 4D object, the user can observe the object from arbitrary directions in the 4D space. The details of this relation between the head position and the 4D sphere will be described in the next part. The sensors which are attached to both hands are used to define the distance between the 4D viewpoint and the 4D object, and used to determine the 3D eye point. The distance between the right hand's sensor and the left hand's sensor corresponds to the distance between the 4D viewpoint and the 4D object. Thus, we will be far away from the 4D object when we open both hands widely. On the contrary, we will approach the 4D object when we narrow both hands. Moreover, we can enter the inside of the object transparently. The direction of both hands corresponds to the 3D viewpoint. Therefore, we can observe the 3D object from various directions if we change the direction of both hands.

In case of the slice method, the sensor attached to the user's head corresponds to the direction of slice which equals the normal vector of the hyper plane which slices the 4D object. The user can observe the object sliced from arbitrary directions by letting the head position correspond to the point which lies on the surface of the 4D sphere. The details of this correspondence will be also described in the next part. The distance between both hands corresponds to the depth of slice which means the translation of a hyper plane. Thus, if we open and narrow hands, we can observe various shape changes of the sliced object. The direction in which both hands point corresponds to the 3D viewpoint. Therefore, we can observe the 3D object from various directions if we change the direction of both hands.

Display Method Data from Motion Capture	Projection	Slice
3D Position of user's head	4D viewpoint on 4D sphere surrounding the object (E_x, E_y, E_z, E_w)	Direction of Slice Normal vector of the hyper plane (a, b, c, d)
Direction to connect both hands	3D viewpoint to observe the projected 3D object (E_x, E_y, E_z)	3D viewpoint to observe the 3D shape of the cross section (E_x, E_y, E_z)
Distance between both hands	Distance between the 4D viewpoint and the 4D object	Depth of Slice Translation of the hyper plane

Table 3. Relation between cotrols by the motion capture and display methods

5.2 Correspondence between 3D space and a 4D sphere

As mentioned above, we need to make the position of user's head correspond to the point which lies on the 4D sphere. Before considering the 4D case, let us begin with a study on the 3D case.

We will explain the correspondence between the 2D plane and the surface of the 3D sphere. As shown in *Figure 9*, when the user moves distance d from the origin in the direction of (x, y) in the 2D plane, we make it correspond to the movement of distance d from the point $(0, 0, r)$ in the direction of $(x, y, 0)$ along the surface of the 3D sphere. For example, when the user moves distance $2\pi r$ from the origin in the 2D plane, he comes back to the starting point $(0, 0, r)$ in 3D space. By this mapping, we can make the 2D plane correspond to all points of the surface of the 3D sphere. According to the distance from the origin in 2D, the 2D shape projected from 3D changes periodically.

In addition, r is determined in proportion to the distance between both hands. In other words, if the user narrow both hands, he will approach the center of the sphere.

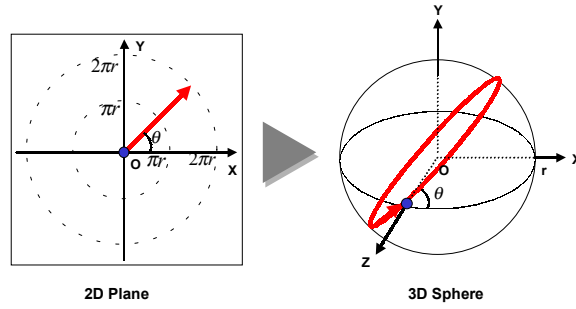


Figure 9. Correspondence between a 2D Plane and a 3D Sphere

Next, we extend this 3D analogy to the 4D case. That is, we consider the correspondence between 3D space and the surface of a 4D sphere. As shown in *Figure 10*, we define 3D space where the user exists and assume the 4D sphere of radius r surrounding the object. In the same way as the 3D case, we define the point $(0, 0, 0, r)$ which lies on the 4D sphere as the origin of 3D space. As shown in *Figure 10*, when the user moves distance d from the origin in the direction of (x, y, z) in 3D space, we make it correspond to the movement of distance d in the direction of $(x, y, z, 0)$ from the point $(0, 0, 0, r)$ along the surface of the 4D sphere. For example, when the user moves distance $2\pi r$ from the origin in 3D space, he comes back to the starting point $(0, 0, 0, r)$ in the 4D space. Equation (3) summarizes this correspondence. By these equations, we can map 3D space onto the surface of a 4D sphere.

In the same way as the 3D case, r is determined in proportion to the distance between both hands. In other words, if the user reduces the distance between both hands, he will approach the center of the object. Moreover, the user can enter even the inside of the 4D object.

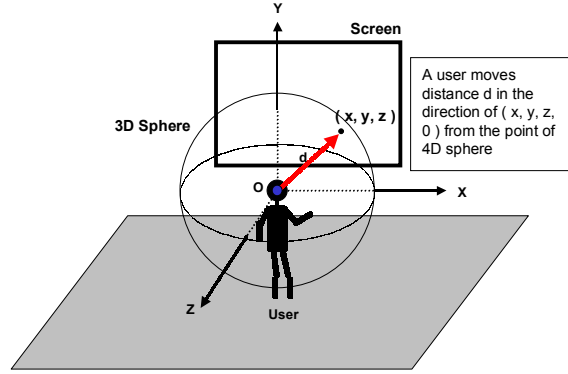


Figure 10. Correspondence between 3D Space and the 4D Sphere

$$\begin{cases} X = \sin(r) \cdot \frac{x}{r} \\ Y = \sin(r) \cdot \frac{y}{r} \\ Z = \sin(r) \cdot \frac{z}{r} \\ W = \sqrt{1 - (X^2 + Y^2 + Z^2)} \end{cases} \quad (3)$$

where $r = \sqrt{x^2 + y^2 + z^2}$

However, an important problem still remains. Even if a 4D viewpoint is decided, we can not uniquely project the 4D object onto 3D space without determining the up vector of viewing volume. If the up vector is not defined, even the same 3D shape shows a variety of 3D postures. To solve this problem we define this up vector according to the path which the user traced. That is, we calculate the up vector by incrementing the differential rotation according to the movement of the user. This method enable us to observe a periodical and continuous change of an object. Note that the 3D posture of a 4D object does not only depend on the user's 3D position but also depends on the user's path.

6 EXAMPLES OF 4D DATA DISPLAY

4D data can be classified into four types according to the relationship among the axes. We show the following examples of the four types of data displayed by the proposed two methods.

1. 4 - 0 data e.g. 4D solid object: 4D cube
2. 3 - 1 data e.g. 3D scalar function: $w = \exp\{-(x^2 + y^2 + z^2)\}$
3. 2 - 2 data e.g. complex function: $f(z) = 1/z$
4. 1 - 3 data e.g. 3D time-series data : 3D moving image

Note that the user observes stereoscopic shape images which change interactively according to his motion, although the images given here are 2D still ones.

6.1 The Projection Method

An image of a 4D cube by the projection method is shown in *Figure 14*. We draw each volume, hyper surface of the 4D cube, by altering colors. We also draw each edge as silver to recognize the 3D shape easily. According to user's motion in 3D space, the 3D appearance of the 4D cube changes.

An image of 3D Gaussian function is shown in *Figure 15*. We draw the x, y, z and w axes with red, green, blue and yellow, respectively. The hue of the surface corresponds to the w value. The maximum point can be found easily by alteration of the view point.

An image of complex function $1/z$ is shown in *Figure 16*. The input complex plane corresponds to (x, y) and the output complex plane corresponds to (z, w). We alter the color of the graph from blue into yellow as the value w changes from positive to negative. The singular point can be found easily in the origin.

An image of time-series motion of the human model is shown in *Figure 17*. The model raises and lowers both hands. The hue of the object corresponds to time.

We show an example of the 4D simple clipping in *Figure 18*. A user observes a 4D rotation of 4D cube at the inside of a 4D cube.

6.2 The Slice Method

An animation of the section of a 4D cube is shown in *Figure 19*. The direction of slice is (1, 1, 1, 1). The hue of the image corresponds to the depth of slice. The section alters its shape from a tetrahedron into a octahedron and also returns to a tetrahedron as the depth of slice translates.

An animation of the section of 3D Gaussian function is shown in *Figure 20*. The direction of slice is (1, 1, 1, 1). The section becomes a 2D surface in this case. The maximum can be found in the center of the section.

An animation of the section of complex function $1/z$ is shown in *Figure 21*. The direction of slice is (1, 1, 1, 1). The section becomes a 1D line in this case. The detail of the function around the origin can be observed by the variation of the section.

An animation of the section of the time-series motion is shown in *Figure 22*. The direction of slice is (0, 0, 0, 1).

6.3 Interactive Environment

Sequential images that a user experiences with our VR system are presented in *Figure 23-26*.

Figure 23 shows that the user's 3D position corresponds to the point which lies on the surface of a 4D sphere surrounding a 4D object in the projection method. According to user's 3D motion, the 4D cube alters its 3D shape. *Figure 24* shows that the direction of both hands corresponds to the 3D viewpoint.

Figure 25 shows that the user's 3D position corresponds to the point which lies on the surface of a 4D sphere in the slice method. According to user's motion, the 3D shape of section changes. *Figure 26* shows that the distance between both hands corresponds to the depth of slice in the slice method. The color of the object corresponds to the depth of slice.

7 EXPERIMENTS

To examine the effectiveness of our system and to compare two display methods, we conducted quantitative evaluation experiments. For the quantitative evaluation, we measured time and accuracy of a particular task. A 3D analogy of the task is shown in *Figure 11*. The user starts from a random initial position around the object. When we observe the cube from the front of each face, it looks like a square. In the 4D case, when we observe a 4D cube from the front of each hyper face, it looks like a 3D cube. The task is that the user searches for a point where a 4D cube is observed like a 3D cube. We measure time and accuracy from the start to the goal. The accuracy is determined by the dot product between the current vector and the goal vector. It is desirable for time to be shortened as repeating the task. It is also desirable for accuracy to be higher as the task is repeated.

Experiments were conducted under the following conditions:

- Subjects were 5 men.
- Tasks were attempted 5 times for each display method.

Average scores of experiments are shown in *Figure 12* and *Figure 13*. In *Figure 13*, the accuracy means an average score of dot products over one task. Regarding the evaluation of accuracy, neither method shows the progress of accuracy so much. However, regarding the evaluation of time, the projection method shows good progress of understanding.

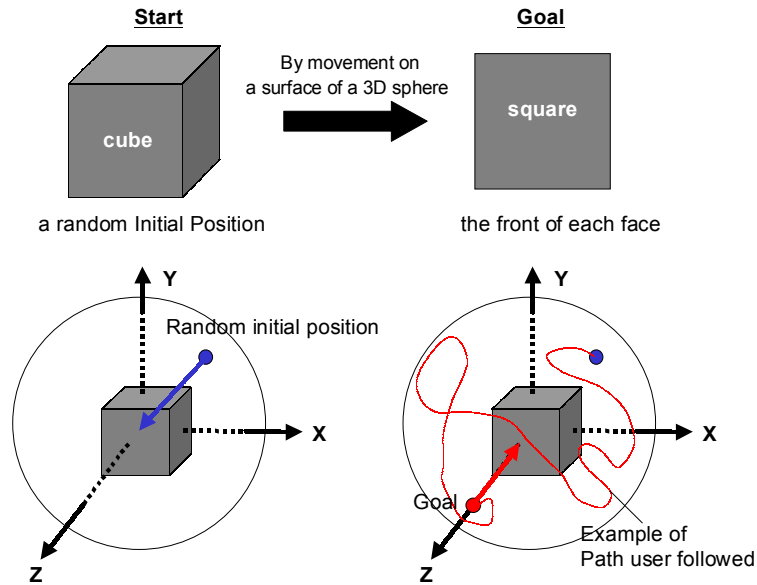


Figure 11. 3D analogy of evaluation experiments

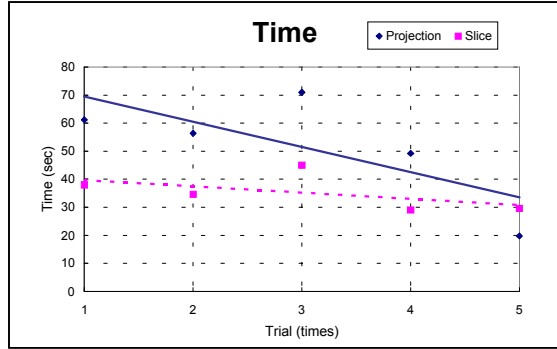


Figure 12. Time of the task

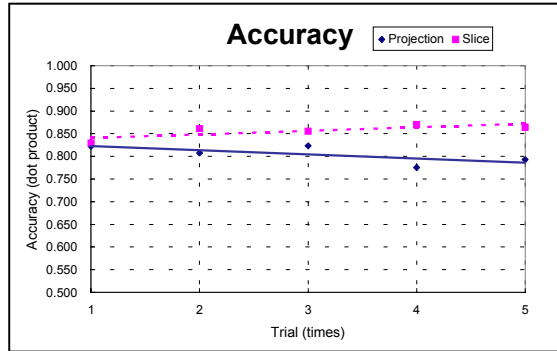


Figure 13. Accuracy of the task

8 CONCLUSIONS

We presented a new virtual reality system for intuitive understanding of 4D data, using a large screen, a stereo viewing system and a motion capture. By using two methods which are the projection and the slice method, we displayed four types of 4D data which can be classified concerning the dependency of each variable. We showed the correspondence between a 3D user's position and the point which lies on the surface of a 4D sphere surrounding a 4D object. With this correspondence, our system enabled the user to observe the 3D shape of the projection and the section from arbitrary points and arbitrary directions in 4D space interactively. Using this system, the user can understand the features of 4D data such as the shape of 4D solid objects, the profile of a 3D scalar field, the singular point of the complex function and the dynamic characteristics of a 3D object motion.

As a result of evaluation experiments, it was shown that this system is effective for

intuitive understanding of 4D objects. Now, we are investigating a training method to adapt the user to 4D space.

4D data often show unexpected visual appearances when represented in the 3D space. Therefore, it is expected that this system is also useful for an artistic application to search and get ideas from multidimensional data.

The described virtual environment will overcome the dimensional limitation on our thinking in order to develop the human imagination and creativity.

References

- [1] T.Kanade "Recovery of the three dimensional shape of an object from a single view" *Artificial Intelligence*, Vol.17, No.1-3, 409-459 1981
- [2] David Hilbert, Stefan Cohn-Vossen "Geometry and Imagination" Chelsea, 1952
- [3] T.Banchoff, J.Werner "Linear Algebra through Geometry" Springer-Verlag 1983
- [4] T.Banchoff "Beyond the Third Dimension : Geometry, Computer Graphics, and Higher Dimensions" Scientific American Library NY 1990
- [5] David Banks "Interactive Display and Manipulation of Two-dimensional Surfaces in Four Dimensional Space" In Symposium on *Interactive 3D Graphics* pp197-207 NY 1992
- [6] David Banks "Illumination in diverse codimensions" In Proceedings of *SIGGRAPH* 1994
- [7] A.J. Hanson. " Geometry for N-dimensional graphics" In Paul Heckbert, editor, *Graphics Gems IV*, pp 149-170. Academic Press, Cambridge, MA, 1994.
- [8] A.J. Hanson. " Rotations for N-dimensional graphics" In Alan Paeth, editor, *Graphics Gems V*, pp 55-64. Academic Press, Cambridge, MA, 1995.
- [9] A.J.Hanson and P.A.Heng "Illuminating the fourth dimension" *IEEE Computer Graphics & Applications*, pp. 54-62 July 1992
- [10] Ikebe Ryoji, Toriwaki Junichiro "A Visualization Method for 4-Dimensional Objects and Data" General Conference of Institute of Electronics, Information and Communication Engineers (IEICE), Japan D-12-114 pp.313 1998
- [11] Mark Philip, Silvio Levy, Tamara Munzner "Geomview: An Interactive Geometry Viewer" available at <ftp://www.geom.umn.edu>
- [12] A.J.Hanson, Konstantine I.Ishkov, Jeff H.Ma "Meshview: Visualizing the Fourth Dimensions" *Course Notes for SIGGRAPH'99 Visualizing Quaternions*, 1999
- [13] Shoemake "A User Interface for specifying three-dimensional orientation using mouse" In Proceedings of *Graphics Interface* pp151-156 May 1992
- [14] A.J.Hanson "The Rolling Ball" In David Kirk editor, *Graphic Gems III* pp51-60 Academic Press 1992

The Projection Method

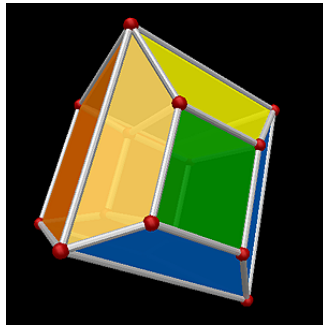


Figure 14. 4D Cube

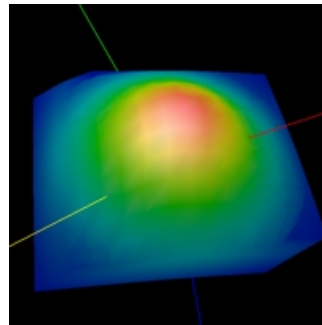


Figure 15. 3D Gaussian Function
 $w = \exp\left[-(x^2 + y^2 + z^2)\right]$

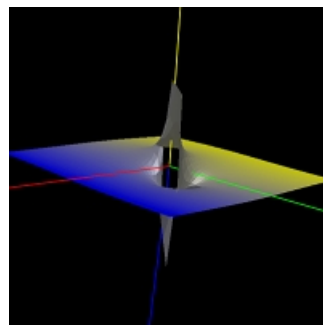


Figure 16. Complex
Function $f(z) = 1/z$

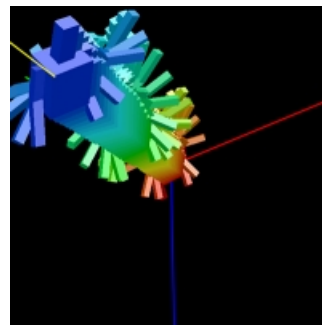


Figure 17. Time-Series
Motion of The Human Model

Inside of the 4D cube

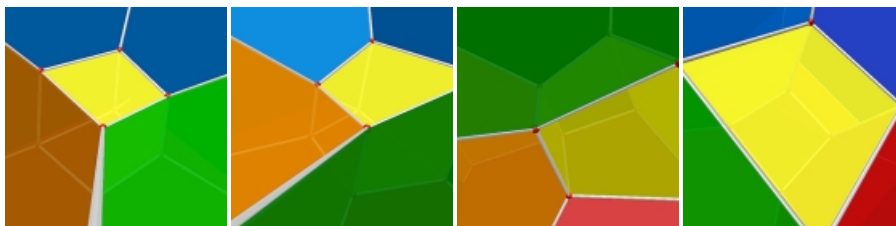


Figure 18. Rotation of 4D cube in the inside of 4D cube

The Slice Method

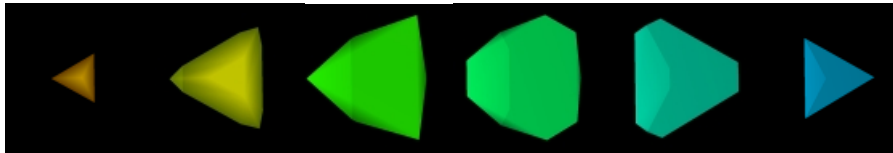


Figure 19. 4D Cube

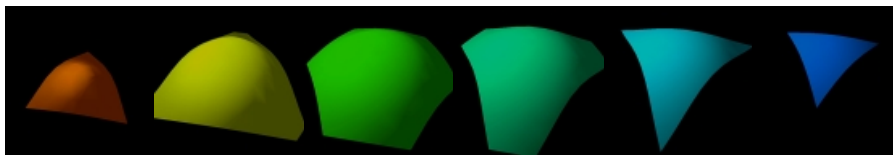


Figure 20. 3D Gaussian Function $w = \exp\{-(x^2 + y^2 + z^2)\}$

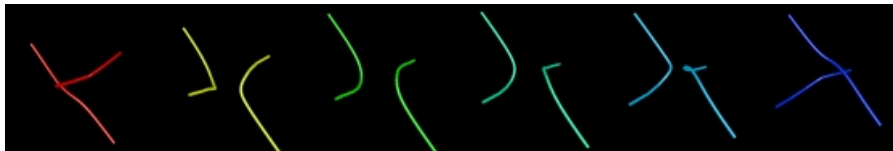


Figure 21. Complex Function $f(z)=1/z$



Figure 22. Time-series Human Motion

Interactive Environment

The Projection Method

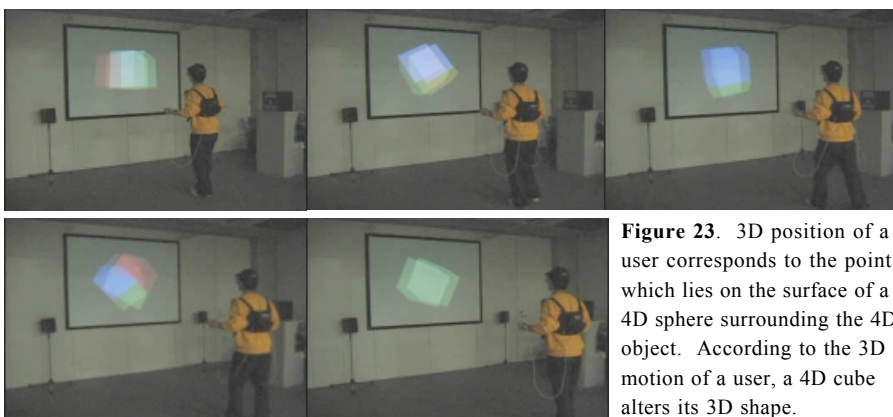


Figure 23. 3D position of a user corresponds to the point which lies on the surface of a 4D sphere surrounding the 4D object. According to the 3D motion of a user, a 4D cube alters its 3D shape.



Figure 24. The direction of both hands corresponds to the 3D viewpoint.

The Slice Method



Figure 25. User's position corresponds to the direction of slice.



Figure 26. The distance between both hands corresponds to the depth of slice. The hue of the object corresponds to the depth of slice.