INTERACTIVE ENVIRONMENT FOR INTUITIVE UNDERSTANDING OF 4D DATA

M. Murata and S. Hashimoto

Humanoid Robotics Institute, Waseda University, Japan

Abstract: We present a new virtual reality system to assist intuitive understanding of four dimensional data. Two mapping methods, which are the projection from 4D space to 3D space and the 3D section of 4D data, are employed to display 3D images generated from 4D data. A user can interactively observe the 4D object from an arbitrary direction by corresponding his 3D position to the point which lies on the surface of the 4D sphere surrounding the object.

Keywords : four dimensions, virtual reality, computer graphics

1 INTRODUCTION

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 have been done in the field of computer vision[1].

As the relationship between 3D space and 2D space can be extended to the higher dimensional case mathematically, it is expected that 3D graphics makes 4D information visible. However, we can not easily understand 4D space.

The difference between the 4D case and the 3D case is very clear. In the 3D case, we already have 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.

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

In this paper, we present a novel system to let everyone understand 4D data. In addition, we introduce interactive techniques which make user's 3D position correspond to the surface of the 4D sphere.

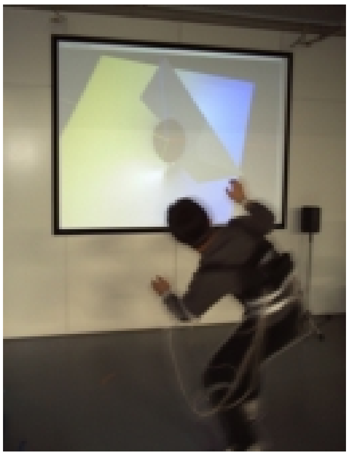


Figure 1. Interactive Environment

2 APPROACH

2.1 Related Works

From the 70's, using 3D computer graphics, studies on displaying 4D data were begun. Using computer graphics, Thomas Banchoff studied the 3D projection and section techniques to display 4D geometry[2][3]. However, the graphics hardware did not have enough performance for the interactive animation at that time.

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

In the 90's, A. J. Hanson proposed several 4D rendering algorithm which were carefully extended from the 3D computer graphics algorithm. He summarized the theory of 4D geometry[6]. He also investigated the 4D light source and visualized a realistic 4D cube and many mathematical objects which reflect 4D shading information by extending 3D algorithm such as the Gouraud and Phong shading[7].

These works have enabled us to observe realistic images of 4D data represented in 3D. However, if the algorithm which needs much computational power is used, the complexity of computation becomes larger in 4D than 3D.

Recently, Geomview and Meshview which visualize mathematical 4D data were proposed[8,9]. By extending studies on user interface for manipulating a 3D object to the 4D case, Geomview and Meshview enabled us to observe 4D mathematical data interactively.

2.2 Research Goals

As mentioned in the previous part, existing approaches to visualize 4D data can be categorized into two ways. The first approach is to create a realistic 3D projection by extending the 3D rendering algorithm. The second approach is to focus on the interaction between a user and 4D data.

We focus on the second approach. We consider that the real-time interaction is more effective to understand 4D space than observation of a realistic still image.

Compared with the related works, our approach is new in the display method and 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 choose the appropriate method to display 4D data. As interface, a motion capture is employed as the input interface so that the user can manipulate 4D objects more intuitively.

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 the 4D world through the interaction in our interactive environment.

3 SYSTEM OVERVIEW

Our system overview is shown in *Figure 2*. We use a motion capture device as an input interface. It is wireless and 6 degree-of-freedom measurement system for multiple points on user's body based on the magnetic field. In addition, this device can work in real-time and transmit data via Ethernet. We use the position data of three sensors which are attached to user's head and both hands. The position data are transmitted to SGI Onyx2 through network.

In Onyx2, the position data of the user's head is mapped to the 4D view point or the direction of slice. Using these information, the system draws the projection or the section of 4D data in real-time. We use OpenGL to draw these images onto the 100 inch single screen. Stereo glasses are used to understand the 3D shape easily.

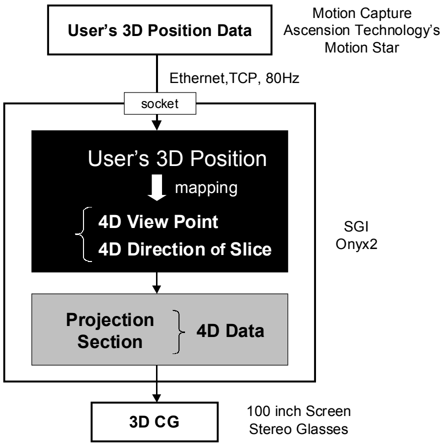


Figure 2. System Overview

4 DISPLAY METHODS

In this section, we will discuss two methods to display 4D data into 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 into 2D space. Our system deals with the 4D Euclidean space where the four axes are perpendicular each other. The fourth axis which is equal with other three axes is not limited to the time axis.

4.1 The Projection Method

The 4D projection method is an extension of 3D rendering algorithm. The projection is natural and powerful method to display 3D objects into a 2D plane.

The flow chart of this method is depicted in *Figure 3*. The Input data to the system are 4D object data and the viewpoint coordinates of a user. The 4D object data include whole vertex coordinates and topological information about each polygon. Using the viewpoint coordinates, we project the 4D vertex coordinates onto 3D space. Topological information about each polygon is conserved after the projection. Therefore, we can draw the 3D shape generated from the 4D data as a binocular stereo image into the screen. To increase a reality of 3D shape, we use a large screen and stereo glasses.

To observe the 4D object from an arbitrary point, our system allows the user to alter the viewpoint in 4D space. As we focus on the real-time interaction between the user and 4D data, we do not consider the present of 4D light source which increases the complexity of computation. We are interested in the interactive operation such as the real-time change of the observational orientation in 4D space.

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

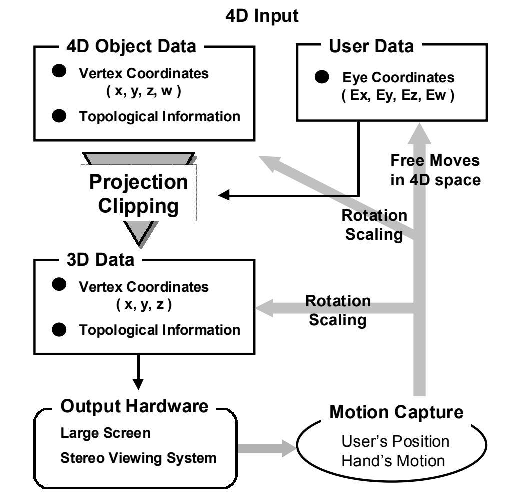


Figure 3. Overview of the Projection Method

Controls	Purpose	User's Motion or Operation	Interface
Move in 4D	To observe a 4D object from the 4D arbitrary point	Walking aroud 3D space	Motion Capture
Rotation in 3D	To understand a 3D shape generated from the 4D object	Altering the direction to connect both hands	Motion Capture
Rotation in 4D	To observe a 4D 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 a 3D shape with appropriate scale	Dragging the mouse	GUI
Drawing Mode	To observe a 4D 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 callb hyper plane in 4D space. The 4D section between a 4D object and the hyper plane generates a 3D solid object in 3D space. According to the translation of the hyper plane in 4D space, the 3D shape of the section shows various transformations. It is expected that this animation of the moving section give us an insight to understand 4D objects.

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

The 4D object data include whole vertex coordinates, topological information about each polygon and hyper plane data which compose the 4D object. The 4D object data includes whole vertex coordinates, topological link information about each polygon and information about volumes which compose the 4D object. Using these input data, we calculate the section of the 4D object and the hyper plane. The Output data is shape data of the section which includes 3D coordinates of all vertex and link information about each 3D polygon. Because the section is a 3D solid object, we draw the 3D section as a binocular stereo image into the screen to understand its shape 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 the hyper plane. Moreover, the user can alter the depth of slice. When the user alters the depth of slice along with a normal vector of the hyper plane, the section shows various kinds of animation. It is expected that these animation images become effective insight 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 is summarized in *Table 2*.

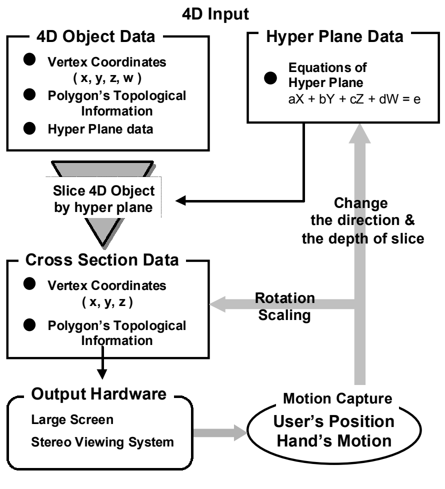

4D Input
4D Object Data
● Vertex Coordinates (x, y, z, w)
● Polygon's Topological Information
● Hyper Plane data
Hyper Plane Data
● Equations of Hyper Plane aX + bY + cZ + dW = e
Slice 4D Object by hyper plane
Change the direction & the depth of slice
Cross Section Data
● Vertex Coordinates (x, y, z)
● Polygon's Topological Information
Rotation Scaling
Output Hardware
Large Screen
Stereo Viewing System
Motion Capture
User's Position
Hand's Motion

Figure 4. Overview of the Slice Method

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	Selecing 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 Data

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

Display Method Data from Motion Capture	**Projection**	**Slice**
3D Position of user's head	4D viewpoint on 4D sphere surrounding the object (Ex, Ey, Ez, Ew)	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 (Ex, Ey, Ez)	3D viewpoint to observe the 3D shape of the cross section (Ex, Ey, Ez)
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 and display methods

5.2 Correspondence Between the 3D Space and the 4D Sphere

As mentioned above, we need to make the 3D position of user's head data 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. That is, we will explain the correspondence between a 2D plane and the surface of a 3D sphere. As shown in *Figure 5*, when a user moves for distance *d* from the origin in the direction of (x, y) in the 2D plane, we make it correspond 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, if a user moves for distance 2 π r from the origin in the 2D plane, he will come back to the starting point (0, 0, r) in 3D space. By this mapping, we can let the 2D plane correspond to all points lies on the surface of the 3D sphere. According to the distance from the origin in 2D, the 2D shape which is projected from 3D changes periodically.

Next, we will extend this 3D analogy to the 4D case. That is, we must consider the correspondence between the 3D cubic space and the surface of a 4D sphere. As shown in *Figure 6*, we define 3D space where the user exists and assume a 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 6*, when the user moves for 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, if the user moves 2π r for distance from the origin in 3D space, he will come back to the starting point (0, 0, 0, r) in 4D space.

However, an important problem still remains. Even if the 4D viewpoint is decided, we can not uniquely project the 4D object onto 3D space without determining the up vector of viewing volume. To solve this problem we define this up vector according to the path the user traced. That is, we calculate the up vector by incrementing the differential rotation according to the movement of the user. This method enables us to observe a periodical and continuous change of an object.

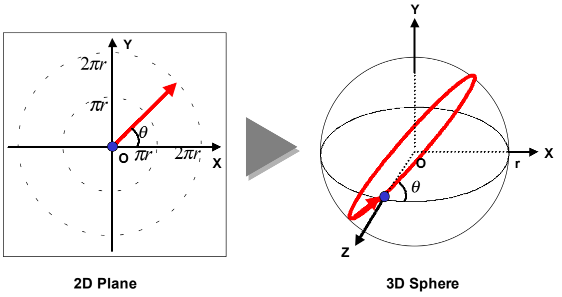
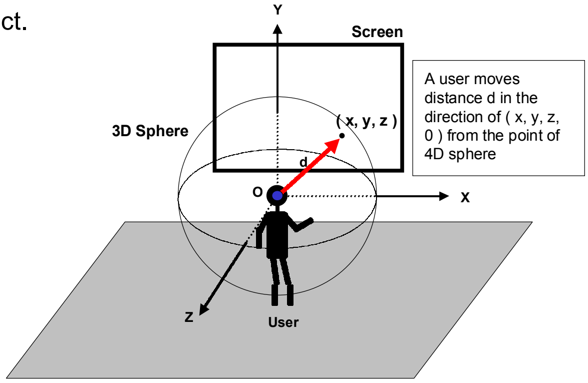


Figure 5. Correspondence between the 2D Plane and the 3D Sphere



A user moves distance d in the direction of (x, y, z, 0) from the point of 4D sphere

Figure 6. Correspondence between 3D Space and the 4D Sphere

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: $w = \exp\left\{ \left(x^2 + y^2 + z^2\right)\right\}$
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.

The above images rendered by the projection method are shown in *Figure 7*, *8*, *9* and *10* respectively. In *Figure 8*, the hue of the surface corresponds to the w value. In *Figure 9*, The input complex plane is (x, y) and the output complex plane is (z, w). We alter the color of the graph from blue into yellow as the value x changes from positive to negative. The singular point can be found easily in the origin. The maximum value, feature and tendency of the 4D data are found by the projected 3D shape.

The above images rendered by the slice method are shown in *Figure 11*, *12*, *13* and *14* respectively. The direction of slice is (1, 1, 1, 1) in *Figure 11*, *12* and *13*. The direction of slice is (0, 0, 0, 1) in *Figure 14*. The hue of the image corresponds to the depth of slice.

Finally, we present sequential images that a user experiences our VR system. *Figure 15* shows that the 4D cube alters its 3D shape according to user's 3D motion. *Figure 16* shows that the distance between both hands corresponds to the scale. *Figure 17* shows that the direction of both hands corresponds to the 3D viewpoint. *Figure 18* shows that the distance between both hands corresponds to the depth of slice. The color of the 3D shape corresponds to the depth of slice.

7 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. Using two methods that are the projection and the slice method, we displayed four types of 4D data which are classified concerning the dependency of each variable. We showed the correspondence between user's 3D position and the point which lies on the surface of a 4D sphere surrounding a 4D object. By this correspondence, our system enabled the user to observe the 3D shape of the projection and the section from an arbitrary point in 4D space.

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

4D data often shows unexpected visual appearances when mapped into 3D space. Therefore, it is also expected that this system becomes useful for artistic application to search and get ideas from multidimensional data.

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] T.Banchoff, J.Werner "Linear Algebra through Geometry" Springer-Verlag 1983

[3] T.Banchoff "Beyond the Third Dimension : Geometry, Computer Graphics, and Higher Dimensions" Scientific American Library NY 1990

[4] David Banks "Interactive Display and Manipulation of Two-dimensional Surfaces in Four Dimensional Space"In Symposium on *Interactive 3D Graphics* pp197-207 NY 1992

[5] David Banks "Illumination in diverse codimensions" In Proceedings of *SIGGRAPH* 1994

[6] A.J. Hanson. "Geometry for N-dimensional graphics" In Paul Heckbert, editor, *Graphics Gems IV,* pp 149-170. Academic Press, Cambridge, MA, 1994.

[7] A.J.Hanson and P.A.Heng "Illuminating the fourth dimension" *IEEE Computer Graphics & Applications*, pp. 54-62 July 1992

[8] Mark Philip, Silvio Levy, Tamara Munzner "Geomview: An Interactive Geometry Viewer" available at ftp://www.geom.umn.edu

[9] A.J.Hanson, Konstantine I.Ishkov, Jeff H.Ma "Meshview: Visualizing the Fourth Dimensions" *Course Notes for SIGGRAPH'99 Visualizing Quaternions*, 1999

AUTHORS: Makoto MURATA, Shuji HASHIMOTO, Humanoid Robotics Institute, Waseda University
17 Kikui-cho, Shinjuku-ku, Tokyo 162-0044, Japan, Phone +81-3-3203-4385, Fax +81-3208-8714
E-mail: { makoto,shuji }@shalab.phys.waseda.ac.jp, URL: http://www.phys.waseda.ac.jp/shalab/

The Projection Method

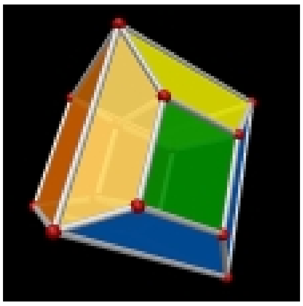


Figure 7. 4D Cube

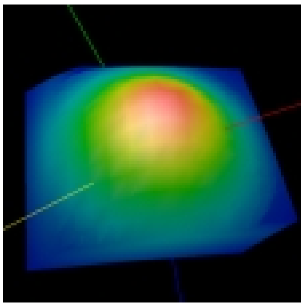


Figure 8. 3D Gaussian $w=\exp\{\ (x^2+y^2+z^2)\}$

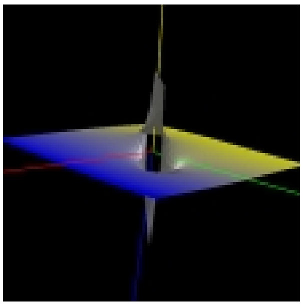


Figure 9. Complex Function *f(z) = 1/z*

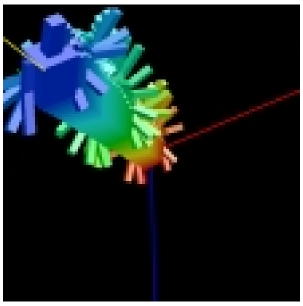


Figure 10.Time-Series Human Motion

The Slice Method

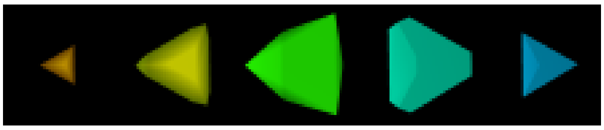


Figure 11. 4D Cube

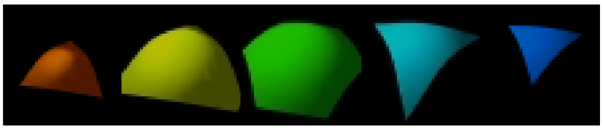


Figure 12. 3D Gaussian $w=\exp\{\ (x^2+y^2+z^2)\}$

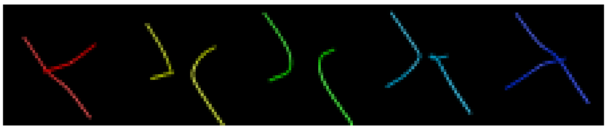


Figure 13. Complex Function *f(z)=1/z*

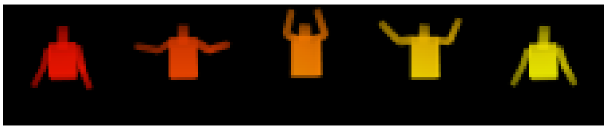


Figure 14. Time-series Human Motion

Interactive Environment

The Projection Method

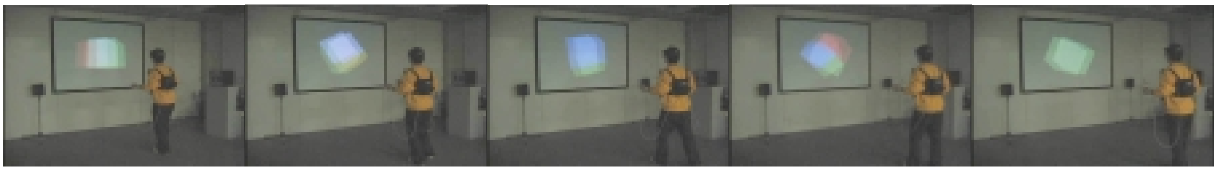


Figure 15. User's 3D position corresponds to the point which lies on the surface of a 4D sphere. According to the 3D motion of a user, a 4D cube alters its 3D shape.

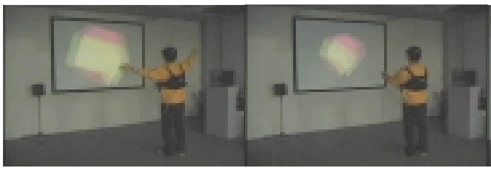


Figure 16. The distance between both hands corresponds to the scale.

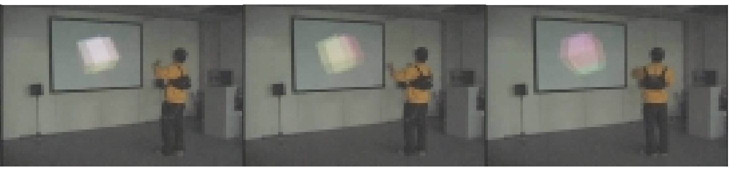


Figure 17. The direction to connect both hands corresponds to the 3D viewpoint.

The Slice Method

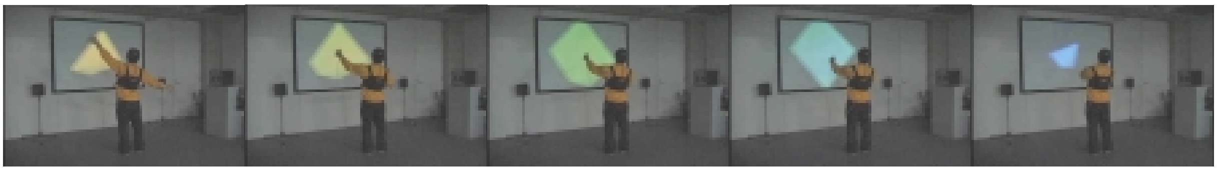


Figure 18. The distance between both hands corresponds to the depth of slice. The hue of the object corresponds to the depth of slice.

NOTICE: All color images in this page are available as a PDF file at the following WWW page. *http://www.phys.waseda.ac.jp/shalab/~makoto/*