Original Paper　*Forma*, **26**, 11–18, 2011

Four-dimensional Mathematical Data Visualization via "Embodied Four-dimensional Space Display System"

Yukihito Sakai[1,2*] and Shuji Hashimoto[3]

[1]*Faculty of Information Sciences and Arts, Toyo University, 2100 Kujirai, Kawagoe, Saitama 350-8585, Japan*
[2]*Research Institute for Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan*
[3]*Faculty of Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan*
E-mail address: yukihito@toki.waseda.jp

(Received February 4, 2011; Accepted April 21, 2011)

In general, it is difficult to visualize 4-D mathematical data with interdependence among variables in 3-D space that we are able to perceive. In our previous work, we developed an embodied 4-D space display system for observing various 4-D objects; the system can intuitively overview and understand the shapes of 4-D objects from various 4-D positions and directions in 4-D space. The system enables us to create any 4-D mathematical data without omitting any properties. Using the system, we can obtain various mapping structures of 4-D mathematical data such as complex functions and functions with three variables, and easily discover characteristics such as the singular point of the complex function and the extremal value of the 3-D scalar function. It is expected that our approach will be useful for the development of perception of 4-D mathematical data.
Key words: 4-D mathematical data, 4-D Euclidean space, 4-D visualization, 4-D interaction

1. Introduction

We draw diagrams of space information and measurement results within a 2-D plane or 3-D space. In primary education, functions and geometry are treated as 2-D data using graph paper; 4-D data such as solid geometry, complex mathematics, and extreme value analysis are taught at the advanced level. Generally, when we represent 4-D mathematical data in 3-D space, one dimension of the 4-D space must be degenerated. In order to understand an overview image of the interdependence among variables, it is desirable for 4-D mathematical data to be visualized inclusively in mathematical fields.

In our previous work, we developed 4-D space display system; this system corresponds to a telescope or a microscope to help understand 4-D information. We can observe various 4-D objects from an arbitrary 4-D eye-point and direction, while moving freely in 4-D space. Our system deals with a 4-D Euclidean space with four axes orthogonal to each other. For 4-D data representation, the system has a framework with geometric and topological information that can visualize arbitrary 4-D data (Sakai and Hashimoto, 2004). We previously reported visualizations such as 4-D solids (regular polytopes) and 3-D time-series data and showed the extensive effectiveness of the system (Sakai and Hashimoto, 2006, 2007a, 2007b). We can provide an interactive environment so that educators and scientists who deal with 4-D data can observe and construct 4-D spaces easily.

In this paper, following on from our previous work, we handle and visualize various 4-D mathematical data in the developed system. We create 4-D mathematical data such as complex functions and functions with three variables. We will be able to discover geometric characteristics such as the singular point of a complex function and the extremal value of a 3-D scalar function. First, related work and our previous work are introduced in Sec. 2. In Sec. 3, we describe how to visualize 4-D data, and explain the outline of interactions in 4-D space. Our system makes it easy to observe 4-D mathematical data. In Sec. 4, we demonstrate some 4-D mathematical data for achieving a new understanding of unusual 3-D perception. Finally, conclusions are given in Sec. 5.

2. Related Work and Our Previous Work

Conventional 4-D visualizations are divided into three categories. The first one performs the perspective projection for 4-D data from the single direction of four axes (Dewdney, 1986; Hausmann and Seidel, 1994; Hanson, 1998). The second one visualizes various 3-D projection drawings of 4-D data from different eye-points in 4-D space (Miyazaki and Ishihara, 1989; Hollasch, 1991). In the former case, the 4-D eye-point is fixed. In the latter case, changes in the 4-D eye-point are limited. The third one slices the 4-D data with a hyperplane in 4-D space (Woodring *et al.*, 2003). In this approach, hyperplane generation is not directly assumed in 4-D space. There are a few reports on interactions with 4-D objects. In existing interactive environments, keyboard input or a joystick associated with human actions are often used for geometric operations of a 4-D object (D'Zmura *et al.*, 2000; Aguilera, 2006); this enable the user to experience the rotation of a 4-D object in 4-D space. However, in these methods, the user's 4-D viewing position is fixed and the user is not allowed to move around the 4-D object in 4-D space.

We constructed an intuitive and interactive 4-D space display system that made human actions in 3-D space corre-

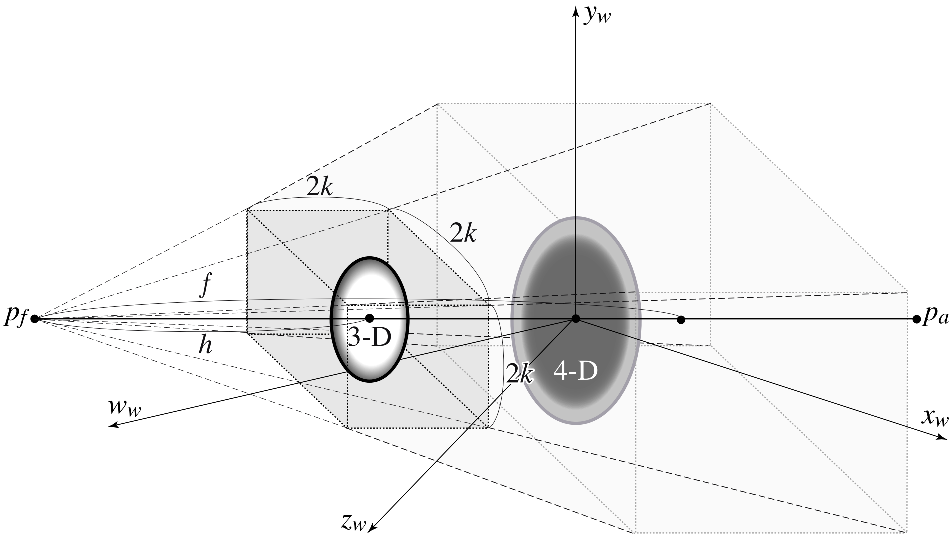


Fig. 1. Visualization model of 4-D space.

spond to movements of the eye-point in 4-D space. The system consists of a glasses-free 3-D display, a flight-controller pad, and a personal computer. For 4-D data representation, the system has a framework with geometric and topological information that can visualize arbitrary 4-D data (Sakai and Hashimoto, 2004). Using the system, we can smoothly observe any 4-D information from various 4-D positions, directions, and distances (Sakai and Hashimoto, 2006). Moreover, our 4-D visualization model can represent various projections, not only the perspective projection, but also the parallel projection and slice operation (Sakai and Hashimoto, 2007a). Using the system to its full extent, we performed various visualizations of 4-D spatio-temporal information to develop a novel understanding of moving 3-D objects in 4-D space-time (Sakai and Hashimoto, 2007b). By conducting evaluation experiments, we showed the effectiveness of the system and obtained an understanding of various 4-D data. Most users realized how the user interface works without a detailed explanation, and intuitively achieved some new insights from observational results in 4-D space.

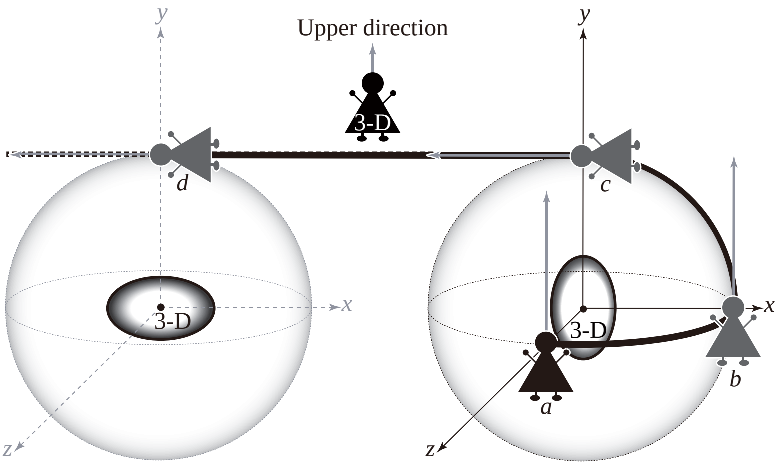
Although some attempts at 4-D visualization and 4-D interaction have been reported, our system differs in the generality of the 4-D data processing and in having a user interface that allows real-time 4-D space travel. The embodied 4-D space display system corresponds to a telescope or a microscope for observing and understanding arbitrary 4-D data. Following on from our previous work, this paper discusses the visualization of 4-D information in general using the developed system.

3. Interactive 4-D Space Display System for Intuitive Usability

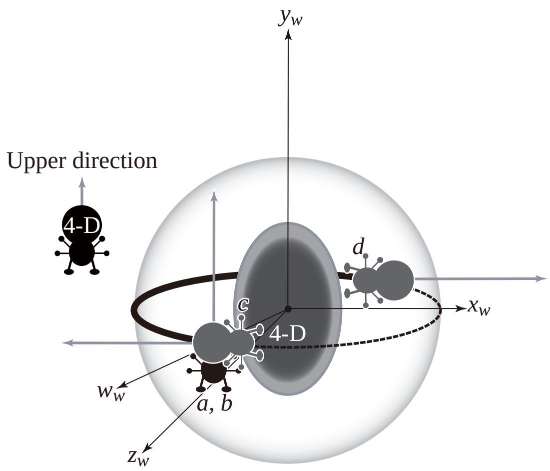
Using our system, we can freely walk in 4-D space and overview 4-D data with an arbitrary 4-D viewing-field (Sakai and Hashimoto, 2006, 2007a). In this section, we outline a visualization algorithm for 4-D data, and explain the interaction method in 4-D space.

As shown in Fig. 1, 4-D data are displayed on a 3-D screen to steer the visual axis in various directions from the eye-point (the look-from point) p_f $(x_{p_f}, y_{p_f}, z_{p_f}, w_{p_f})$ to the observed point (the look-at point) p_a $(x_{p_a}, y_{p_a}, z_{p_a}, w_{p_a})$ in the 4-D world-coordinate system $x_w y_w z_w w_w$. The center of the 3-D screen and the center of the background hyperplane are located on the 4-D visual axis at distances h and f $(> h)$ from the 4-D eye-point, respectively. The volume of the 3-D screen is $2k \times 2k \times 2k$. The 4-D viewing-field is defined as a truncated pyramid that is formed by the 4-D eye-point, the 3-D screen, and the background hyperplane. The 4-D visualization algorithm is composed of a viewing-field transformation, a perspective transformation, a display transformation, and a clipping operation in 4-D space. This framework enables one to observe a variety of 4-D information from an arbitrary 4-D eye-point and visual axis. Furthermore, this 4-D visualization model can represent various projections, not only the perspective projection, but also the parallel projection and the slice operation, by controlling the 4-D viewing-field. The parallel projection is performed by locating the 4-D eye-point extremely far from the 3-D screen. This operation approximates the size of the background hyperplane to that of the 3-D screen. Slice operation is realized by locating the 3-D screen near the background hyperplane. In a similar fashion to that in 3-D space, we can observe various 3-D projection drawings of a 4-D object from an arbitrary 4-D eye-point and visual axis, while changing the form of the 4-D viewing-field to understand the 4-D space.

We had some difficulties moving around in 4-D space to interact with 4-D data because actions in 3-D space are not directly related to movements in 4-D space. In order to overcome such difficulties, we developed a 4-D movement algorithm that makes human actions in 3-D space correspond to movements and rotations on a 4-D spherical surface (see Fig. 2). For example, as shown in Fig. 2(a), movements on paths ab and bc along the 3-D spherical surface correspond to rotations ab and bc at a point on the 4-D spherical surface, and thus coincide with changes in the movement direction on the 4-D spherical surface (see



(a) Movements of the 3-D person in 3-D space



(b) Rotations and movements of the 4-D person on a 4-D spherical surface

Fig. 2. Relations between actions in 3-D space and actions on a 4-D spherical surface.

Fig. 2(b)); the movement of path *cd* in the upper direction in 3-D space, as shown in Fig. 2(a), corresponds to movement on path *cd* along the 4-D spherical surface (see Fig. 2(b)). This method can perform various 4-D actions, not only the above-mentioned eye-point changes on the 4-D spherical surface, but also visual axis changes in 4-D space and movements in the direction of the 4-D object from the 4-D eye-point.

As shown in Fig. 3, we constructed a real-time environment using intuitive interfaces such as a glasses-free 3-D display and a controller pad like a flight simulator. This environment made human actions in 3-D space correspond to movements of the eye-point in 4-D space. Using the system, we can smoothly observe any 4-D data from the desired positions, directions, and distances in 4-D space, thereby extending the experiences of 3-D space. For example, the steering actions of the flight-controller pad are associated with movements on the 3-D spherical surface. In these actions, we can overview a 3-D projection drawing of a 4-D object on the 3-D screen, while rotating at a position on the 4-D spherical surface. The right button on the handle is associated with the movement in the 3-D upper direction in 3-D space. We can observe various 3-D projection drawings of a 4-D object, while moving along the 4-D spherical surface. We can perceive the motion bounded on the 4-D spherical surface through the motion in the upper direction in 3-D space. Using the flight-controller pad, we can perform various 4-D actions, not only the above-mentioned eye-point changes on the 4-D spherical surface, but also visual axis changes in 4-D space and movements in the direction of the 4-D object from the 4-D eye-point. By repeatedly training the system's handling, we will gradually acquire information about positions in 4-D space and a knowledge of 4-D information through interactions in 4-D space.

4. Visualizations of 4-D Mathematical Data

With the interactive system mentioned above, we can smoothly observe any 4-D mathematical data from the desired positions, directions, and distances in 4-D space. In this section, using our interactive environment, we visualize 4-D mathematical data, such as complex functions and functions with three variables, in 4-D space. We attempt to acquire knowledge from interactions with 4-D mathematical data.

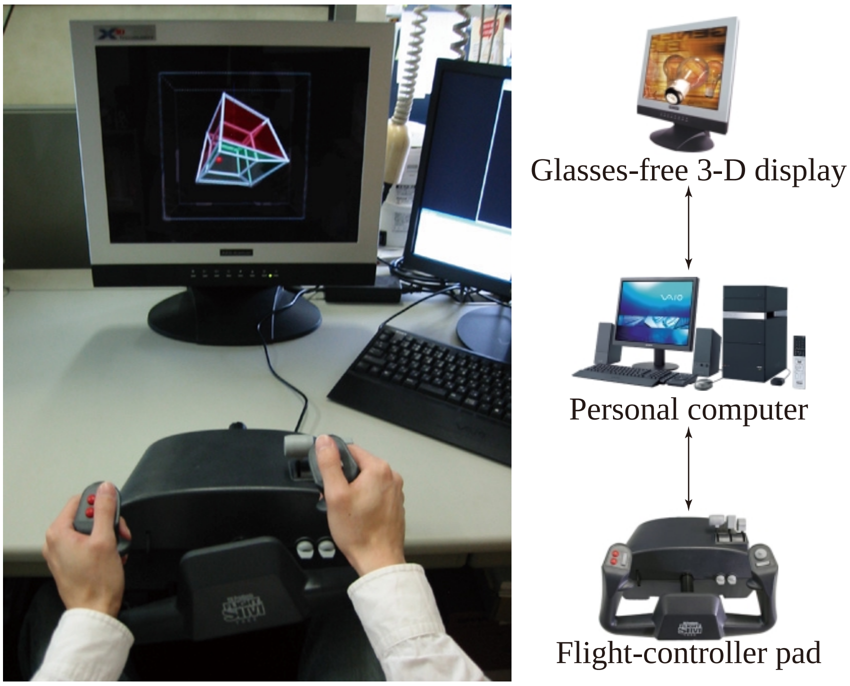


Fig. 3. Interactive 4-D space display system.

4.1 Parameters used for 4-D visualizations

As described in Sec. 3, the parameters p_f, p_a, k, h, and f are the eye-point (the look-from point), the observed point (the look-at point), the scale of the 3-D screen, the distance from the eye-point to the 3-D screen, and the distance from the eye-point to the background hyperplane, respectively (see Fig. 1). The developed system can perform various representations such as perspective projection, parallel projection, and slice operations by controlling the parameters. Parallel projection is performed by locating the eye-point extremely far from the 3-D screen. The scale of the background hyperplane approximately corresponds to that of the 3-D screen. The slice operation is realized by locating the 3-D screen near the background hyperplane. The distance from the eye-point to the background hyperplane becomes approximately congruent with that from the eye-point to the 3-D screen. With the parameter changes mentioned above, we can freely overview 4-D mathematical data with an arbitrary 4-D viewing-field. In the following, we report observational results of various complex functions and functions with three variables.

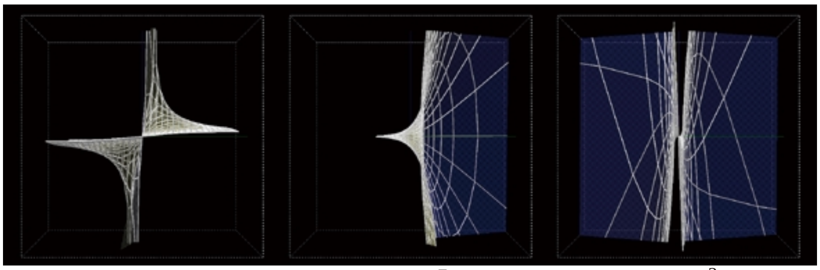
4.2 Visualizations of complex functions

The complex function $f(Z)$ expresses a mapping from a complex plane (x, y) to another complex plane (z, w). The input complex variables $Z = x + iy$ correspond to (x, y), and the output complex variables $f(Z) = z(x, y) + iw(x, y)$ correspond to (z, w). The complex function has a relationship between two inputs and two outputs, and is defined as 4-D information $(x, y, z(x, y), w(x, y))$. Here, the input complex plane (x, y) is assumed to be in the range $-\pi \leq x, y \leq +\pi$. We alter the color of the input complex plane (x, y) from blue to yellow in the direction of the y_w-axis from positive to negative. The parameters p_f, p_a, k, h, and f used for 4-D visualization are given at the bottom of each 3-D image in the following figures.

Figure 4 shows complex functions from various 4-D eye-points by perspective projection. In our system, as the input complex plane is transformed to various shapes by a complex function, we can observe mapping structures of various complex functions in 4-D space. As shown in Figs. 4(a), (b), and (c), we observe the complex functions from eye-points on the w_w-axis in 4-D space. By walking freely in 4-D space, we can perceive the singular point of the complex function, the mapping from a plane to a circle, and the imaginary region of the cubic function (see Figs. 4(d), (e), and (f)). By regarding a complex function as a 4-D object, we can have visual contact with the original shapes to a great extent, and easily understand the geometric structures and characteristics of complex functions.

Figure 5 shows the local structures of complex functions from adjacent 4-D eye-points by perspective projection. When the 4-D eye-point approaches the target of a complex function, we can also search and discover the local structure of the complex function. As shown in Figs. 5(a) and (b), we can observe various 3-D images of complex functions, not only regions near real and imaginary roots, but also chaotic regions (non-convergence regions produced by the Newton method). As shown in Fig. 5(c), we can focus around the singular point only. By using the interactions to freely change the 4-D eye-point, the system helps users to intuitively understand the local characteristics of complex functions.

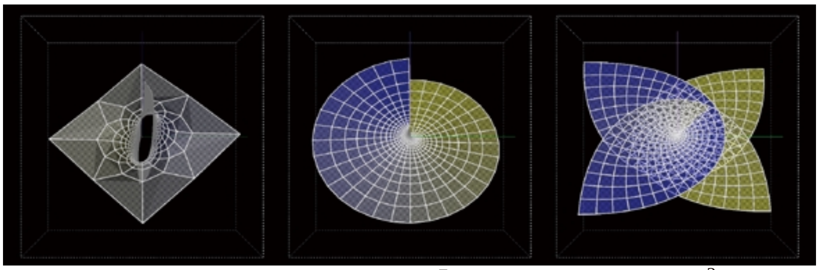
Parallel projection is performed by locating the 4-D eye-point extremely far from the 3-D screen. As one dimension of the 4-D data is degenerated in this operation, we can observe the local area of the complex function. Figure 6 shows mapping structures of complex functions from different 4-D eye-points by parallel projection. As shown in Figs. 6(a),



$f(Z)=1/Z$ (a) $p_f\,(0, 0, 0, 4)$, $p_a\,(0, 0, 0, 0)$, k 0.5, h 0.5, f 200

$f(Z)=e^Z$ (b) $p_f\,(0, 0, 0, 10)$, $p_a\,(0, 0, 0, 0)$, k 0.5, h 0.5, f 200

$f(Z)=Z^3$ (c) $p_f\,(0, 0, 0, 30)$, $p_a\,(0, 0, 0, 0)$, k 0.5, h 0.5, f 200

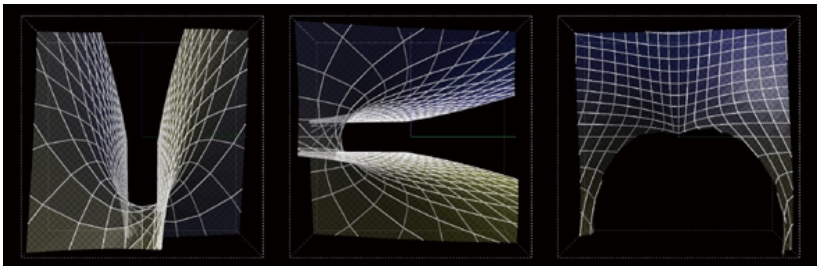


$f(Z)=1/Z$ (d) $p_f\,(4, 4, 0, 0)$, $p_a\,(0, 0, 0, 0)$, k 0.5, h 0.5, f 200

$f(Z)=e^Z$ (e) $p_f\,(20, 20, 0, 0)$, $p_a\,(0, 0, 0, 0)$, k 0.5, h 0.5, f 200

$f(Z)=Z^3$ (f) $p_f\,(50, 50, 0, 0)$, $p_a\,(0, 0, 0, 0)$, k 0.5, h 0.5, f 200

Fig. 4. Mapping structures of complex functions by perspective projection.



$f(Z)=Z^2$-1 (a) $p_f\,(0, 0, 0, 1.5)$, $p_a\,(0, 0, 0, 0)$, k 0.5, h 0.5, f 200

$f(Z)=Z^2$+1 (b) $p_f\,(0, 0, 0, 1.5)$, $p_a\,(0, 0, 0, 0)$, k 0.5, h 0.5, f 200

$f(Z)=1/Z$ (c) $p_f\,(0, 0, 0, 1.5)$, $p_a\,(0, 0, 0, 0)$, k 0.5, h 0.5, f 200

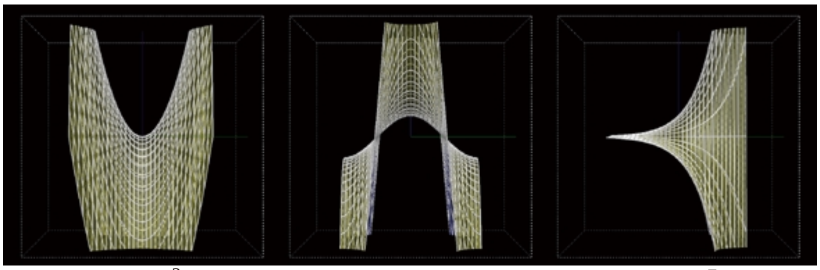
Fig. 5. Local structures of complex functions by perspective projection.

(b), and (c), the input complex plane and the real part of the output complex plane are displayed on the 3-D screen. Figures 6(d), (e), and (f) show the input complex plane and the imaginary part of the output complex plane. This representation in 3-D space is useful for building up organized knowledge from 4-D data.

A slice operation is realized by locating the 3-D screen near the background hyperplane. Figure 7 shows mapping structures and roots of complex functions clipped by a slice operation. In this representation, the section becomes a 1-D line. When we observe the complex function from the 4-D eye-point on the w_w-axis in the 4-D space by a slice operation, we can see the original real function, including the imaginary part, on the 3-D screen (see Figs. 7(a), (b), and (c)). Although the real and imaginary parts of an arbitrary equation are not generally represented in 2-D space, we can have visual contact with the real and imaginary parts of an arbitrary equation in the system at the same time. As indicated by the white circles in Figs. 7(d), (e), and (f), by focusing on the input complex plane, we can have perfect visual contact with not only the real and imaginary roots, but also the conjugate roots, without using numerical mathematics such as quadratic formulas and Newton methods.

4.3 Visualizations of functions with three variables

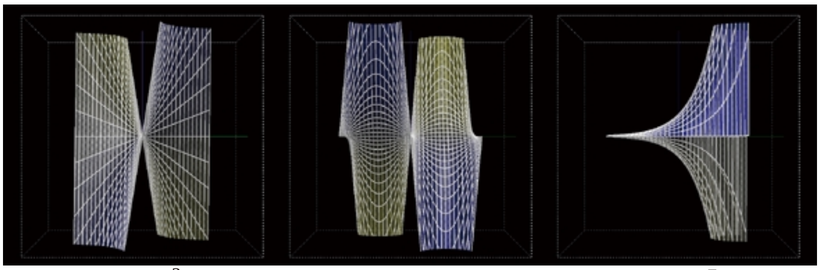
If there are scalar values in a 3-D space, the 3-D scalar function $f(G)$ is defined as the relationship between the three inputs $G = (x, y, z)$ and the one output $f(G) = w(x, y, z)$. For example, the 3-D Gaussian function $f(G) = w(x, y, z) = \exp\{-(x^2 + y^2 + z^2)\}$ is a 3-D scalar potential, and corresponds to the 4-D information



$f(Z)=Z^2$
(a) $p_f(0, 0, 0, 1000)$, $p_a(0, 0, 0, 0)$, k 0.5, h 100, f 3000

$f(Z)=cosZ$
(b) $p_f(0, 0, 0, 1000)$, $p_a(0, 0, 0, 0)$, k 0.5, h 100, f 3000

$f(Z)=e^Z$
(c) $p_f(0, 0, 0, 1000)$, $p_a(0, 0, 0, 0)$, k 0.5, h 100, f 3000

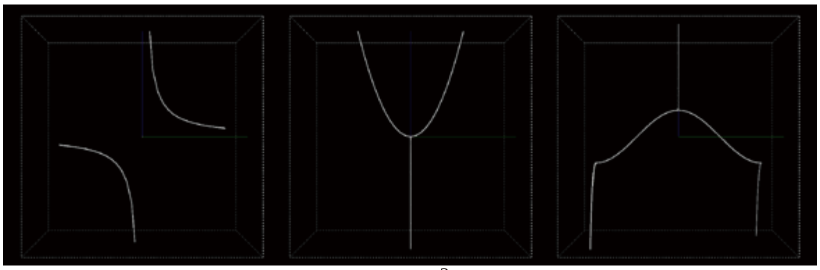


$f(Z)=Z^2$
(d) $p_f(0, 0, 1000, 0)$, $p_a(0, 0, 0, 0)$, k 0.5, h 100, f 3000

$f(Z)=cosZ$
(e) $p_f(0, 0, 1000, 0)$, $p_a(0, 0, 0, 0)$, k 0.5, h 100, f 3000

$f(Z)=e^Z$
(f) $p_f(0, 0, 1000, 0)$, $p_a(0, 0, 0, 0)$, k 0.5, h 100, f 3000

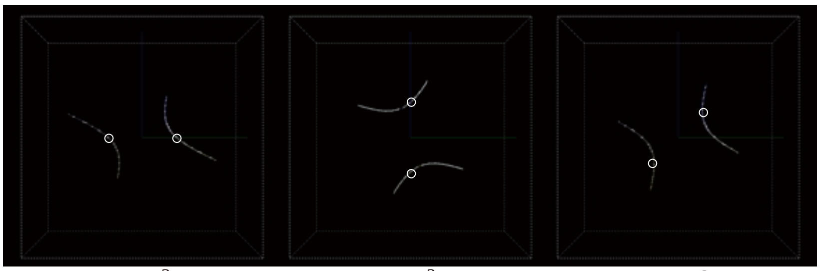
Fig. 6. Mapping structures of complex functions by parallel projection.



$f(Z)=1/Z$
(a) $p_f(0, 0, 0, 1.5)$, $p_a(0, 0, 0, 0)$, k 4, h 1.5, f 1.51

$f(Z)=Z^2$
(b) $p_f(0, 0, 0, 1.5)$, $p_a(0, 0, 0, 0)$, k 4, h 1.5, f 1.51

$f(Z)=cosZ$
(c) $p_f(0, 0, 0, 1.5)$, $p_a(0, 0, 0, 0)$, k 4, h 1.5, f 1.51

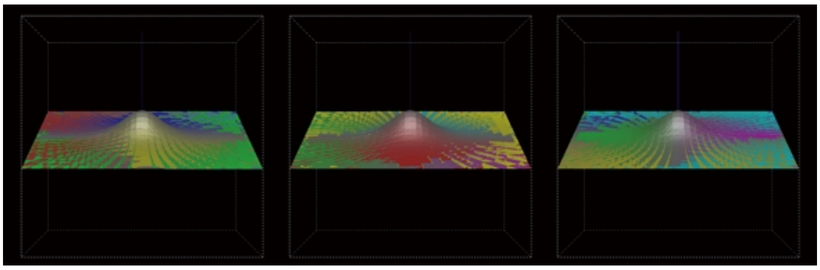


$f(Z)=Z^2$-1
(d) $p_f(0, 0, 1, 1)$, $p_a(0, 0, 0, 0)$, k 3, h 1.41, f 1.45

$f(Z)=Z^2$+1
(e) $p_f(0, 0, 1, 1)$, $p_a(0, 0, 0, 0)$, k 3, h 1.41, f 1.7

$f(Z)=Z^2-i$
(f) $p_f(0, 0, 1, 1)$, $p_a(0, 0, 0, 0)$, k 3, h 1.41, f 1.45

Fig. 7. Mapping structures and the roots of complex functions by slice operations.

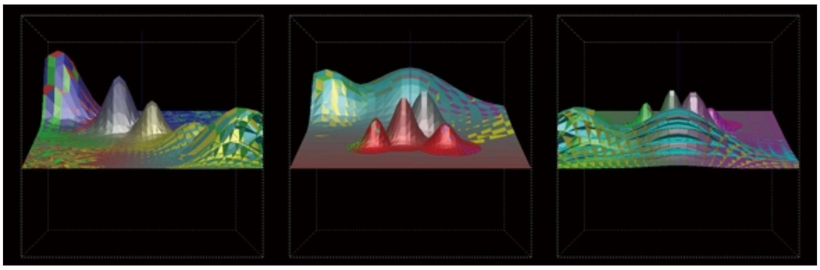


3-D Gaussian function

(a) p_f(-4, 0, 0, 0), p_a(0, 0, 0, 0), k 0.5, h 0.5, f 200

(b) p_f(0, -4, 0, 0), p_a(0, 0, 0, 0), k 0.5, h 0.5, f 200

(c) p_f(0, 0, -4, 0), p_a(0, 0, 0, 0), k 0.5, h 0.5, f 200



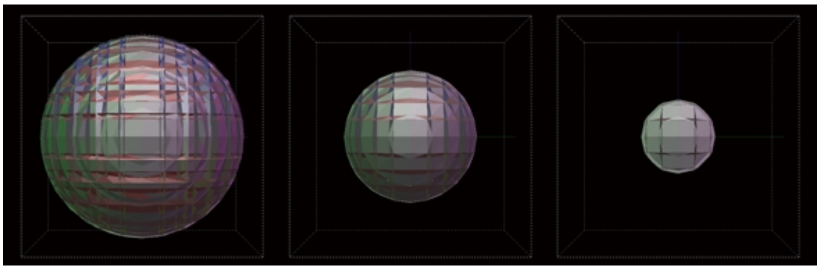
3-D Gaussian function with some peaks

(d) p_f(-4, 0, 0, 0), p_a(0, 0, 0, 0), k 0.5, h 0.5, f 200

(e) p_f(0, -4, 0, 0), p_a(0, 0, 0, 0), k 0.5, h 0.5, f 200

(f) p_f(0, 0, -4, 0), p_a(0, 0, 0, 0), k 0.5, h 0.5, f 200

Fig. 8. Mapping structures of 3-D Gaussians by perspective projection.

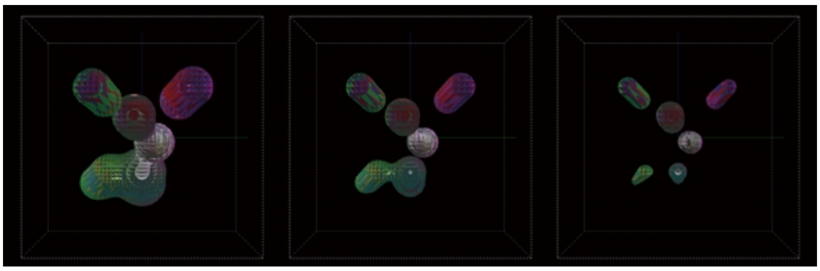


3-D Gaussian function

(a) p_f(0, 0, 0, 1), p_a(0, 0, 0, 0), k 1, h 0.6, f 0.8

(b) p_f(0, 0, 0, 1.3), p_a(0, 0, 0, 0), k 1, h 0.6, f 0.8

(c) p_f(0, 0, 0, 1.6), p_a(0, 0, 0, 0), k 1, h 0.6, f 0.8



3-D Gaussian function with some peaks

(d) p_f(0, 0, 0, 1), p_a(0, 0, 0, 0), k 4, h 0.6, f 0.8

(e) p_f(0, 0, 0, 1.3), p_a(0, 0, 0, 0), k 4, h 0.6, f 0.8

(f) p_f(0, 0, 0, 1.6), p_a(0, 0, 0, 0), k 4, h 0.6, f 0.8

Fig. 9. Extremal values of 3-D Gaussian functions clipped by a slice operation.

$(x, y, z, w(x, y, z))$. Here, the range of the input variables (x, y, z) is assumed to be $-\pi \le x, y, z \le +\pi$. The 3-D Gaussian function is visualized by changing the color, depending on the values of the x_w-, y_w-, and z_w-axes. The parameters p_f, p_a, k, h, and f used for 4-D visualization are given at the bottom of each 3-D image in the following figures.

Figure 8 shows 3-D Gaussian functions seen from the 4-D eye-point on the x_w-, y_w-, and z_w-axes. As we do in viewing a contour map, we can understand the relation between 2-D space and scalar values (see Figs. 8(a), (b), and (c)). The maximum point can be found easily by alteration of the eye-point in 4-D space. The relief of the surface corresponds to the coordinates of the w_w-axis. As shown in Figs. 8(d), (e), and (f), we can easily discover the number of maximum points. The relationship among several maximum points is visualized at various positions and areas on the 3-D screen.

Figure 9 shows the 3-D Gaussian functions clipped by a slice operation. When moving in the positive direction of the w_w-axis, we can take a wide view of the changing area of the maximum points. In this case, the section becomes a 2-D surface. This surface corresponds to the isosurface. As shown in Figs. 9(a), (b), and (c), the maximum point can be found spherically in the center of the section on the 3-D screen. As shown in Figs. 9(d), (e), and (f), some maximum points are visualized in some positions of the section on the 3-D screen. We can observe the same level part of some peaks. By using the slice operation, we can have visual contact with the various critical points of the 3-D Gaussian functions such as the maximum point, the saddle point, and the minimum point, as they topologically change on the 3-D screen.

As mentioned above, the developed system can comprehensively deal with 4-D data $(x, y, z, w(x, y, z))$, including both the position in 3-D space and the scalar value. In general volume-rendering techniques, although some scalar values are searching and the isosurface is rendered, the system can visualize the isosurface on the 3-D screen by a single slice operation. We therefore do not need to implement the visualization algorithm for isosurface generation. Unlike conventional isosurfacing techniques, the system provides a unified framework that enables various 4-D visualizations, not only isosurfaces by slice operations, but also global images by perspective projection and local images by parallel projection.

5. Conclusions

We performed various visualizations of 4-D mathematical data to develop an understanding of complex functions and functions with three variables in 4-D space. In our environment, we were able to observe 4-D mathematical data flexibly from an arbitrary 4-D position and direction by controlling the 4-D viewing-field. We therefore had visual contact with not only mapping structures and roots of complex functions but also the isosurfaces of 3-D Gaussian functions that cannot generally be seen in 3-D space. It is expected that our approach will be useful for developing perceptions of 4-D mathematical data.

Acknowledgments. This work was partially supported by a Grant-in-Aid from the WABOT-HOUSE Project, Gifu Prefecture; the CREST project, "Foundation of technology supporting the creation of digital media contents," of the Japan Science and Technology Agency (JST); and the Global COE (Centers of Excellence) Program, "Global Robot Academia," of the Japan Society for the Promotion of Science (JSPS).

References

Aguilera, J. C. (2006) Virtual reality and the unfolding of higher dimensions, in *Proceedings of SPIE, Stereoscopic Displays and Virtual Reality Systems XIII*, **6055**, pp. 605–612.

Dewdney, A. K. (1986) Computer recreations: A program for rotating hypercubes induces four-dimensional dementia, *Scientific American*, 8–13.

D'Zmura, M., Colantoni, P. and Seyranian, G. D. (2000) Virtual environments with four or more spatial dimensions, *Presence*, **9**(6), 616–631.

Hanson, A. J. (1998) Computer graphics beyond the third dimension, in *Course Notes for SIGGRAPH'98*, Course 46.

Hausmann, B. and Seidel, H.-P. (1994) Visualization of regular polytopes in three and four dimensions, *Computer Graphics Forum*, **13**(3), 305–316.

Hollasch, S. R. (1991) Four-space visualization of 4D objects, M.S. Dissertation, Arizona State University.

Miyazaki, K. and Ishihara, K. (1989) *Four-Dimensional Graphics*, Asakura-Shoten Publishers, Tokyo (in Japanese).

Sakai, Y. and Hashimoto, S. (2004) 4D solid representation for visualization, in *Proceedings of the 10th International Conference on VIRTUAL SYSTEMS and MULTIMEDIA—VSMM 2004*, pp. 593–601.

Sakai, Y. and Hashimoto, S. (2006) Interactive four-dimensional space visualization using five-dimensional homogeneous processing for intuitive understanding, *The Journal of the Institute of Image Information and Television Engineers*, **60**(10), (108)1630–(125)1647.

Sakai, Y. and Hashimoto, S. (2007a) Four-dimensional space visualization with four-dimensional viewing field control, *Bulletin of the Society for Science on Form*, **21**(3), 274–284 (in Japanese).

Sakai, Y. and Hashimoto, S. (2007b) Four-dimensional space-time visualization for understanding three-dimensional motion, *The Journal of the Institute of Image Electronics Engineers of Japan*, **36**(4), 371–381.

Woodring, J., Wang, C. and Shen, H.-W. (2003) High dimensional direct rendering of time-varying volumetric data, in *Proceedings of the 14th IEEE Visualization, 2003*, pp. 417–424.