



A visualization method of four-dimensional polytopes by oval display of parallel hyperplane slices

Kageyama, Akira

(Citation)

Journal of Visualization, 19(3):417-422

(Issue Date)

2016-08

(Resource Type)

journal article

(Version)

Accepted Manuscript

(Rights)

©The Visualization Society of Japan 2015. The final publication is available at Springer via <http://dx.doi.org/10.1007/s12650-015-0319-5>

(URL)

<https://hdl.handle.net/20.500.14094/90004022>



A Visualization Method of Four Dimensional Polytopes by Oval Display of Parallel Hyperplane Slices

Akira Kageyama

Received: date / Accepted: date

Abstract A method to visualize polytopes in a four dimensional euclidian space (x, y, z, w) is proposed. A polytope is sliced by multiple hyperplanes that are parallel each other and separated by uniform intervals. Since the hyperplanes are perpendicular to the w axis, the resulting multiple slices appear in the three-dimensional (x, y, z) space and they are shown by the standard computer graphics. The polytope is rotated extrinsically in the four dimensional space by means of a simple input method based on keyboard typings. The multiple slices are placed on a parabola curve in the three-dimensional world coordinates. The slices in a view window form an oval appearance. Both the simple and the double rotations in the four dimensional space are applied to the polytope. All slices synchronously change their shapes when a rotation is applied to the polytope. The compact display in the oval of many slices with the help of quick rotations facilitate a grasp of the four dimensional configuration of the polytope.

Keywords Polytopes · Four dimension · Four dimensional visualization

1 Introduction

In famous book “Flatland”, an inhabitant called Square living in a flat 2-dimensional (2-D) space meets a stranger, Sphere, living in a 3-D space [1]. Mr. Square in fact meets Sphere’s cross section, which is a circle in Flatland. We apply in this paper Mr. Square’s experience to comprehend polytopes in a 4-D euclidian space (x, y, z, w) .

One of the earliest studies on visualization of a 4-D polytope is by Noll in 1968 [2], in which a hypercube is visualized by a perspective projection in 4-D. The resulting 3-D objects, a cube-within-a-cube, is drawn with the wireframe graphics. When the hypercube is rotated in 4-D, an intriguing turning-inside-out motion is observed. The inner cube grows until it becomes the outer cube while the outer cube shrinks until it becomes the inner cube. This motion was filmed as an animation.

Since the work of Noll, the projection method is the major approach to the visualization of 4-D polytopes. The approach is not restricted to the computer graphics. Physical models are built for the projected 4-D regular polytopes with ball-and-sticks [3]. Mathematics of the 4-D regular polytopes and their graphics with Mathematica is found in [4].

MeshView [5] is a program to draw 4-D objects by perspective and orthogonal projections in 4-D with a special emphasis on a 4-D rolling ball interface. Input devices for 4-D rotations and translations are studied in detail in [6]. More advanced control of 4-D transformations is developed by making use of a head tracking system [7], in which a 3-D motion of the user’s head is linked to a view point motion in 4-D space. Various rendering methods have been studied for projected objects in 4-D [8–11] Some of them are applied to data visualization of 4-D scalar field [12, 13].

A very different approach to the visualization of 4-D objects is slicings by a hyperplane in 4-D. In contrast to the projection method, studies on the slicing approach is rather sparse. Hausmann and Seidel developed a program to visualize the six regular polytopes in 4-D [14]. Although their program supports both the projection and the hyperplane slicing approaches, the slice is very restrictive: The hyperplane is always along one of the symmetry axes of each polytope. A visualization of 4-D object by the slicing with an arbitrary hyperplane was investigated by Woodring et al. [13]. Since the purpose of their study is to visualize time varying, three-dimensional scalar field, the w axis corresponds to time and the 3-D slices were visualized by the 3-D volume rendering.

We propose in this paper a 4-D visualization method of polytopes in a 4-D (x, y, z, w) space by taking multiple slices by parallel hyperplanes. The basic idea and procedure are as follows: The slicing hyperplanes are intentionally restricted

A. Kageyama
Department of Computational Science, Kobe University, Kobe 657-8501, Japan
E-mail: kage@port.kobe-u.ac.jp

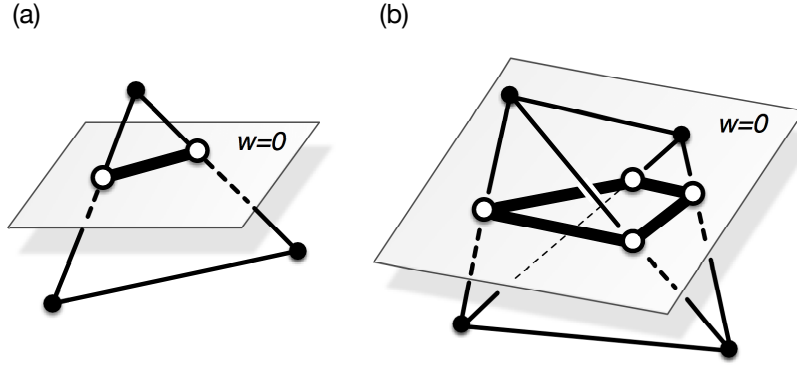


Fig. 1 (a) A cross section of a triangle in 4-D by a hyperplane $w = 0$. The white points are cross sections of two edges of the triangle. The line connecting the two points are a cross section of the triangle. (b) A cross section of a regular tetrahedron by $w = 0$. The quadrilateral with white vertices is a cross section of the tetrahedron by the hyperplane.

to $w = 0$ and its parallels. ($w = \text{constants.}$) Instead of considering every hyperplanes, i.e., $c_x x + c_y y + c_z z + c_w w = c_0$, where c_x etc. are constants, we rotate the polytope in 4-D. The rotation is designed to be applied extrinsically, i.e., the user observe the object under a rest frame and apply the 4-D rotations under the fixed coordinates. The extrinsic rotation with fixed slicing hyperplanes enables the user to keep track of the sense of orientation in 4-D space. The slicing hyperplanes are aligned along the w axis with uniform intervals. It is like the computed tomography applied to 4-D. The resulting multiple slices, which are 3-D objects, are presented in a single window by the standard 3-D computer graphics. An interactive 4-D rotation invoked by the user causes synchronized deformations of all the 3-D slices.

One of the most interesting applications of the 4-D visualization is the scientific visualization. The computational fluid dynamics (CFD), for example, produces field data that are defined in 3-D spatial and 1-D temporal dimensions. When a characteristic spatial structure in a fluid, such as a localized vortex line, exhibits an intriguing dynamical deformation in time, researchers would try to understand the vortex dynamics through the visualization of the data fields in the 4-D space. An approach to the 4-D data visualization is the projection-based method that is represented by Noll's work, in which the field data are integrated in a specific direction to reduce the dimension. This approach is appropriate for grasping a whole picture of the 4-D data, or for observing macroscopic structures of the spatio-temporal coherency of the data. In contrast to that, our approach to the 4-D data visualization proposed in this paper is appropriate for observing microscopic structures of the spatio-temporal coherency in detail, because the numerical data on specific slice planes are explicitly visualized in our method.

2 A Pentachoron and its Hyperplane Slices

Although the method proposed in this paper is a general framework applicable to any 4-D object, we use it for the visualization of 4-D polytopes, especially a regular pentachoron.

A regular pentachoron has three regular tetrahedra around each edge. A slice of a regular pentachoron in 4-D with a hyperplane is a 3-D object that is composed of vertices, edges, and faces in 3-D. Here we visualize them with OpenGL. Consider a hyperplane $w = 0$. The slice of an edge in 4-D by the hyperplane is a vertex in 3-D. Figure 1(a) schematically shows a triangle in 4-D that is sliced by the hyperplane $w = 0$. The two white points are slices of two edges in the triangle. They will be visualized by balls in 3-D with OpenGL. The thick line connecting them is a slice of the triangle. It will be shown as a bar in 3-D.

A slice of a tetrahedron in the regular pentachoron is either a triangle or a quadrilateral. An example of a quadrilateral cross section is shown in Fig. 1(b). The four white points constitute a quadrilateral. The 3-D coordinates of these four points are already calculated when the vertices (balls) and edges (bars) are visualized. Note that the four points are on a plane in 3-D because they are on two hyperplanes in 4-D. The quadrilateral is drawn as a semitransparent face in 3-D. Examples of 3-D slices of a 4-D regular pentachoron are shown in Fig. 2. The panel (a) of this figure is a 3-D slice of a regular pentachoron by $w = 0$ hyperplane, viewed in the negative z direction. The RGB-colored axes in the upper left in each panel of Fig. 2 depict the x , y , and z directions, respectively. The colored vertical lines the bottom in each panel will be explained in the next section.

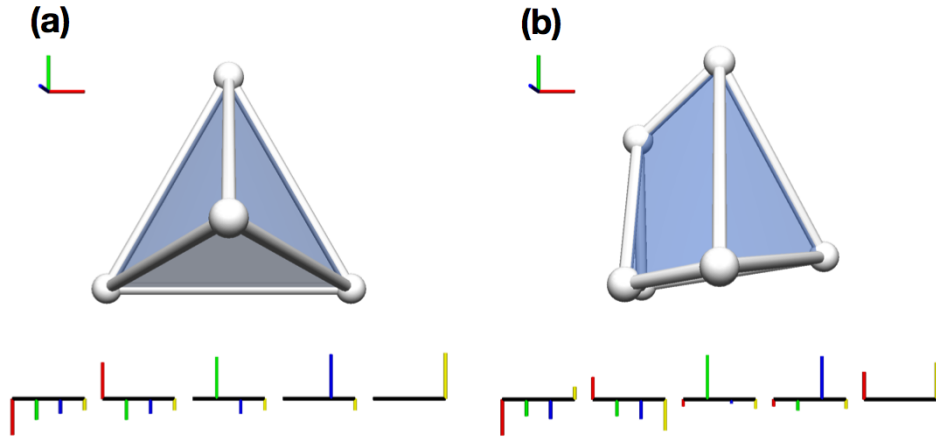


Fig. 2 3-D slices of a 4-D regular pentachoron with a hyperplane. A slice is composed of vertices, edges, and faces that are visualized by balls, bars, and semi-transparent planes. (a) A slice with the hyperplane $w = 0$ of a regular pentachoron before applying any rotation. The 3-D axes in the upper left depict x (red), y (green), and z (blue) directions. The colored vertical lines at the bottom are parallel coordinates of the five vertices of the regular pentachoron. Details are described in the text. (b) A slice of the regular pentachoron by the same hyperplane after a rotation is applied. The change of the five vertices' positions are apparent in the parallel coordinates.

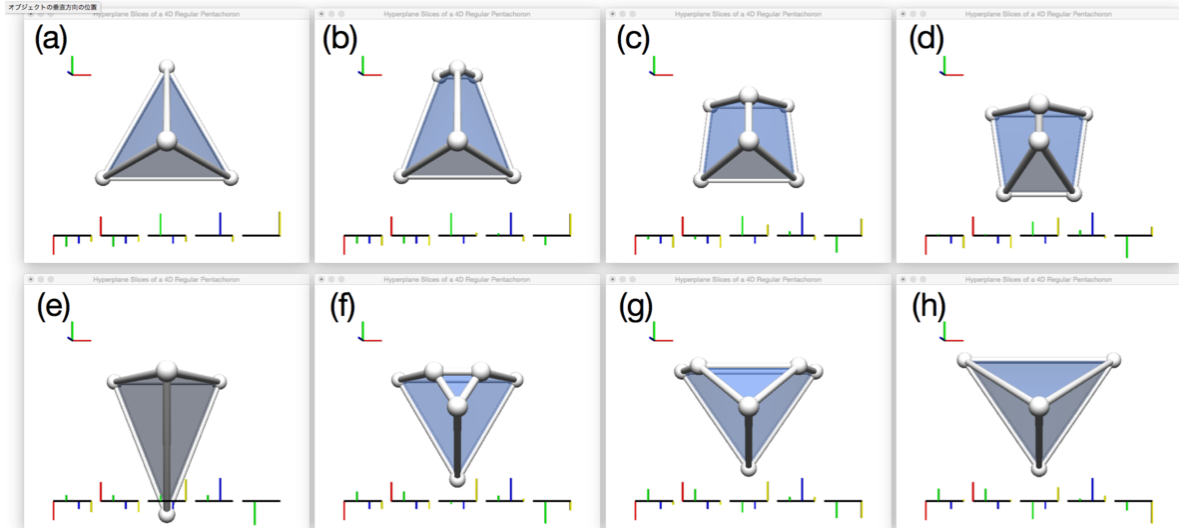


Fig. 3 A sequence of a simple rotation in 4-D. The slice is a hyperplane $w = 0$. The regular pentachoron is rotated in the y - w plane for π radian from (a) to (g). The colored parallel coordinates of the five vertices at the bottom of each panel indicate that y coordinates (green) and w coordinates (yellow) are mixed as the rotations goes on, while other coordinates, x (red), z (blue) remain the same.

3 Multiple Slices and Their Oval Display

The edge length of the regular pentachoron, that is presented in Fig. 2 and other figures in this paper, is 2. The initial condition of the regular pentachoron is shown in Fig. 2(a). The coordinates (x, y, z, w) of the five vertices P_0 to P_4 in the initial condition are as follows: $P_0 : (-1, -1/\sqrt{3}, -1/\sqrt{6}, -1/\sqrt{10})$, $P_1 : (1, -1/\sqrt{3}, -1/\sqrt{6}, -1/\sqrt{10})$, $P_2 : (0, 2/\sqrt{3}, -1/\sqrt{6}, -1/\sqrt{10})$, $P_3 : (0, 0, \sqrt{3}/\sqrt{2}, -1/\sqrt{10})$, and $P_4 : (0, 0, 0, 4/\sqrt{10})$. The center of gravity is on the origin. The vertical short lines at the bottom of Fig. 2 are parallel coordinates for x (red), y (green), z (blue) and w (yellow), of the five vertices P_0 to P_4 . They are placed from left to right in the panel. Figure 2(b) shows a slice with the same $w = 0$ plane of the regular pentachoron that is rotated in 4-D.

A rotation in 4-D is represented by a so-called double rotation. A double rotation is a combination of two simple rotations around two fixed planes that are absolutely perpendicular each other [15, 16]. Here we denote a simple rotation in, for example, the x - y plane with rotation angle α as $R_{xy}(\alpha)$. We also denote a double rotation with angles α and β in the x - y and z - w planes, respectively, as $R_{xy,zw}(\alpha, \beta)$. The rotation applied in Fig. 2(b) from the initial condition (a) is double rotation $R_{xw,yz}(\alpha, \beta)$, where $\alpha = -\pi/\sqrt{3}$ and $\beta = \pi/\sqrt{3} - \pi/\sqrt{8}$. The coordinates of the five vertices are shown in the parallel coordinates at the bottom of the panel (b).

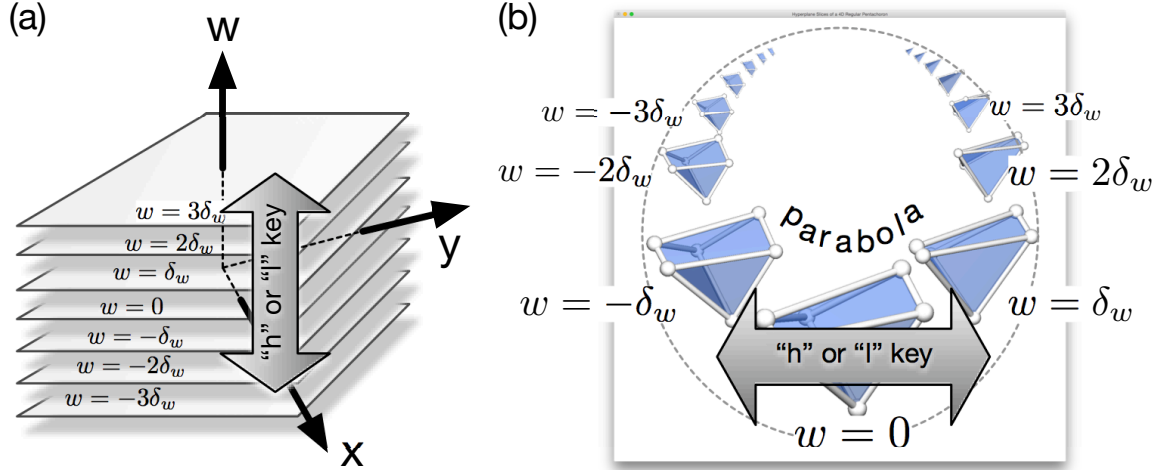


Fig. 4 (a) Multiple slices in 4-D with $w = \text{const.}$ hyperplanes. (b) Oval display of the multiple slices. The 3-D slices are placed on a parabola in the world coordinates in 3-D and viewed by a perspective projection.

To apply rotations (both simple and double) in 4-D, we use the keyboard-based interface proposed in our paper [17]. For example, typing “2” key causes a single rotation $R_{xy}(\alpha)$ and typing “y” key causes a double rotation $R_{xy,zw}(\alpha, \beta)$. The rotation angle α is controlled by the keyboard, too. Typing “k” key (“j” key) increases (decreases) the angle α . This is following the vi-editor’s convention.

Figure 3 shows a sample sequence of a continuously applied simple rotation in the y – w plane, R_{yw} , for π radian in total. The parallel coordinates of the five vertices indicate that y coordinates (green) and w coordinates (yellow) are mixed as the rotations goes on, while other coordinates, x (red), z (blue) remain the same.

While a single slice by a hyperplane conveys very limited information of the target’s 4-D structure, presenting multiple slices, at once, by parallel hyperplanes will facilitate a grasp of the four dimensional configuration as a whole.

Presenting many images that are taken with different parameters is not new in the visualization of higher dimension. For example, HyperSlice [18] is a visualization method for N (> 2) dimensional scalar field. In their method, a point of focus $c = (c_0, c_1, \dots, c_{N-1})$ is placed in the N dimensional scalar field. A 2-D field $S_{ij}(x_i, x_j)$ is defined for each i and j ($0 \leq i \neq j \leq N$) as $f(c_0, \dots, c_{i-1}, x_i, c_{i+1}, \dots, c_{j-1}, x_j, c_{j+1}, \dots, c_{N-1})$. By applying standard visualization methods for 2-D, each field S_{ij} is converted into an image \hat{S}_{ij} . In the HyperSlice method, all the images \hat{S}_{ij} are placed in a matrix form. Comparing whole images in a sight, it is expected that one can grasp the N dimensional structure of the scalar field f .

In our method, we take multiple slices of a 4-D polytope with hyperplanes of $w = 0, \pm\delta_w, \pm2\delta_w, \pm3\delta_w, \dots$, as schematically shown in Fig. 4(a). In order to present as many slices as possible to the viewer, we place them in a parabola curve on a plane in a 3-D world coordinates, as shown in Fig. 4(b). The slices appear in an oval configuration since a parabola looks as an ellipse in the usual (3-D) projection. In the default state, the slice by the hyperplane $w = 0$ is shown at the center. Other slices of $w = n\delta_w$ ($n \geq 1$) are placed in the n -th neighbor in the right. Similarly, the slices of negative w are aligned in the left.

The viewer can change the default value of w that is shown at the center by the keyboard typing. When the “l” key is typed, the default w shifts to $w = +\delta_w$. Repeated typing of the “l” key continuously shifts the central w value. As a consequence, in the oval display, a typing of the “l” key effectively shifts the “focus slice”, which is shown in the center, to the right neighbor. This is again following the vi-editor’s convention. Conversely, a type of the “h” key causes the shift of the “focus slice” to the left.

A rotation applied to the 4-D object leads to synchronous deformations of the slices. The viewer can observe a kind of 4-D tomography of the object, interactively applying both the simple and the double rotations in the 4-D space through the keyboard. An example of continuously applied 4-D rotation to a 4-D regular pentachoron is shown in Fig. 5. The panels (a) to (f) show a sequence of a double rotation. The applied rotation is $R_{xw,yz}(\alpha, \beta)$. In the sequence of (a) to (f), the rotation is consecutively applied with fixed double angles $\alpha = \pi/8$ and $\beta = \pi/\sqrt{3} - \pi/8$, to move from (a) to (b), then from (b) to (c), and so on. One can observe the x – w and y – z coordinates pairs are mixed in these rotations from the parallel coordinates.

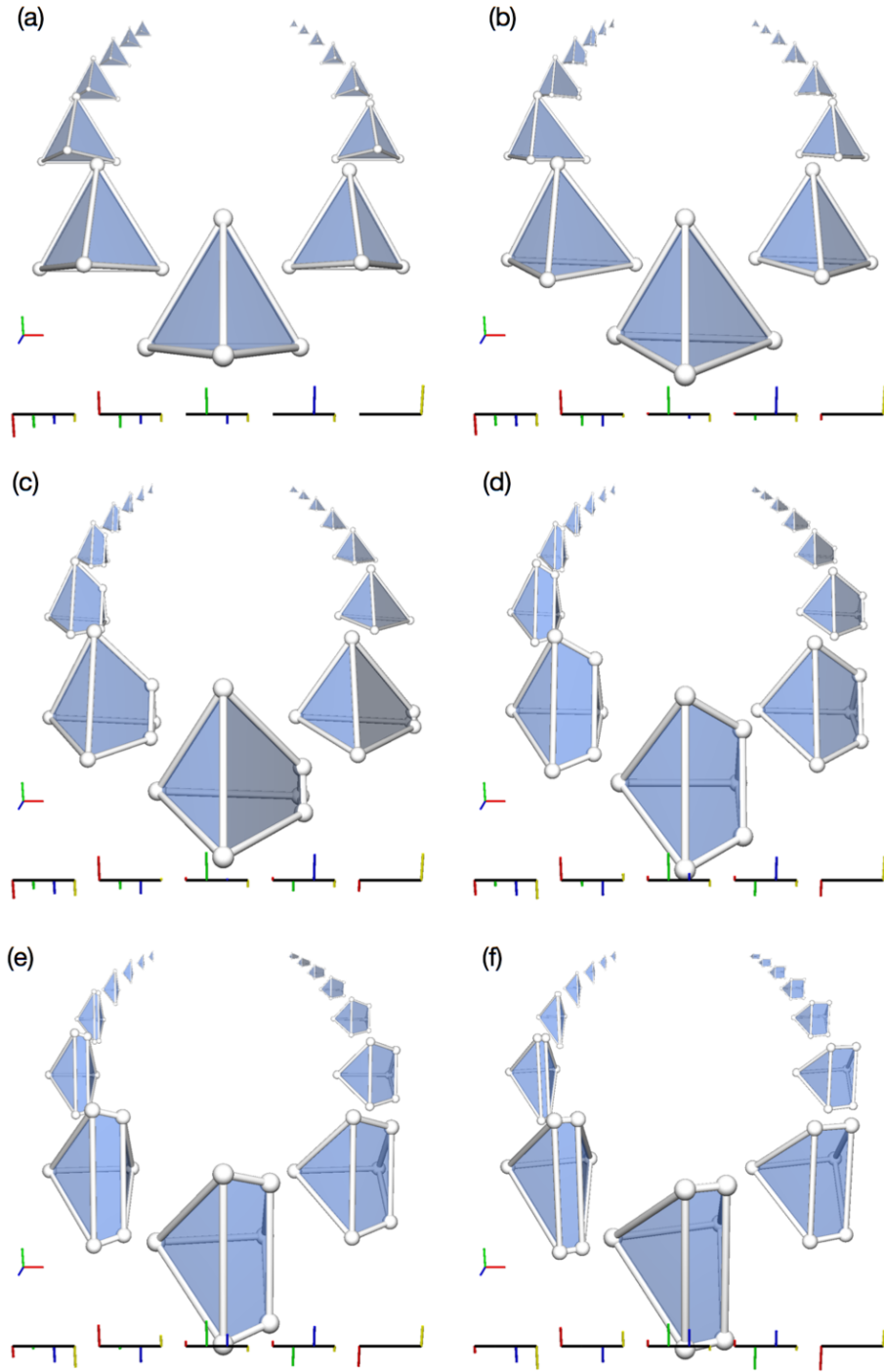


Fig. 5 A time sequence of the oval view of a regular pentachoron. A double rotation $R_{xw,yz}(\alpha, \beta)$ is continuously applied to the regular pentachoron in the panels (a) to (f). The angles are $\alpha = \pi/8$ and $\beta = \pi/\sqrt{3} - \pi/8$.

4 Summary

We have proposed a visualization method of 4-D polytopes. In the proposed method, multiple 3-D slices of a polytope by parallel hyperplanes are presented in oval configuration in a view window. While the hyperplanes are fixed in 4-D, the polytope is interactively rotated in 4-D by the viewer through the key typing of the keyboard. Both the simple and the double rotations are applied by single key typings. The compact view in the oval configuration of many slices facilitates

a grasp of 4-D structure of the polytope. This visualization method is applicable to other 4-D objects than polytopes, especially to the 4-D data visualization produced by CFD.

Acknowledgements This work was supported by JSPS KAKENHI grant No. 20260052. The author thanks a reviewer for pointing out the importance of subsidiary data in the slice visualization. The colored parallel coordinates are added by the reviewer's suggestion.

References

1. E. Abbott, *Flatland : a romance of many dimensions* (Dover Publications, New York, 1992)
2. A.M. Noll, in *Computer animation and the forth dimension*, AFIPS Conference Proceedings, **33** (1968), pp. 1279–1283
3. C.H. Sequin, *3D Visualization Models of the Regular Polytopes in Four and Higher Dimensions*, *Proceedings of Bridges: Mathematical Connections in Art, Music, and Science* (2002), pp. 37–48
4. J.M. Sullivan, *Generating and rendering four-dimensional polytopes*, *THE MATHEMATICA JOURNAL* **1** (1991), pp. 76–85
5. A.J. Hanson, K.I. Ishkov, J.H. Ma, *Meshview: Visualizing the Fourth Dimension*, Technical report (1999), <http://www.cs.indiana.edu/%7Ehansonq/meshview/meshview.pdf> (1999)
6. D. Banks, *Interactive manipulation and display of surfaces in four dimensions*, *Proceedings of the 1992 symposium on Interactive 3D graphics* (1992), pp. 197–207
7. Y. Sakai, S. Hashimoto, *Four-dimensional Mathematical Data Visualization via “Embodied Four-dimensional Space Display System”*, *Forma*, **26**(1), pp. 11–18 (2011)
8. A.J. Hanson, P.A. Heng, *Visualizing the fourth dimension using geometry and light*, *Proceedings of the 2nd conference on Visualization '91* (1991), pp. 321–328
9. C.M. Hoffmann, J. Zhou, *Some techniques for visualizing surfaces in four-dimensional space*, *Computer-Aided Design* **23**(1), pp. 83–91 (1991)
10. A.J. Hanson, R.A. Cross, *Interactive visualization methods for four dimensions*, *Proceedings of Visualization '93* (1993), pp. 196–203.
11. A. Chu, C.W. Fu, A.J. Hanson, P.A. Heng, *GL4D: A GPU-based Architecture for Interactive 4D Visualization*, *IEEE Transactions on Visualization and Computer Graphics*, **15**(6), pp. 1587–1594 (2009).
12. A.J. Hanson, P.A. Heng, *Four-dimensional views of 3D scalar fields*, *Proceedings of Visualization '92* (1992), pp. 84–91.
13. J. Woodring, C. Wang, H.W.W. Shen, *High dimensional direct rendering of time-varying volumetric data*, *Proceedings of VIS 2003* (2003), pp.417–424
14. B. Hausmann, H.P. Seidel, *Visualization of Regular Polytopes in Three and Four Dimensions*, *Computer Graphics Forum* **13**(3), 305–316 (1994).
15. F.N. Cole, *On rotations in space of four dimensions*, *American Journal of Mathematics*, **12**(2), pp. 191–210 (1890)
16. H.P. Manning, *Geometry of Four Dimensions* (MACMILLAN, New York, 1914)
17. A. Kageyama, *Keyboard-based control of four-dimensional rotations*, *J. Vis.*, in press (2015)
18. J.J. van Wijk, R. van Liere, *HyperSlice*, *Proceedings of the 4th conference on Visualization '93* (1993), pp. 119–125