Modeling of Kepler-Poinsot Solid Using Isomorphic Polyhedral Graph

Hidetoshi Nonaka

***Abstract*—**This paper presents an interactive modeling system of uniform polyhedra using the isomorphic graphs. Especially, Kepler-Poinsot solids are formed by modifications of dodecahedron and icosahedron.

***Keywords*—**Kepler-Poinsot solid, Shape modeling, Polyhedral graph, Graph drawing.

I. Intruduction

KEPLER-POINSOT solids are subset of uniform polyhedra, which include 5 regular polyhedra (Platonic soilids), 13 semi-regular polyhedra (Archimedean solids), and 4 regular intersected polyhedra (Kepler-Poinst solids). Traditionally, Kepler-Poinsot solids are formed by stellating or faceting the ordinary dodecahedron and icosahedron, which are regular polyhedra. This paper presents another way to form and to model them using the isomorphic graphs. The system consists of three subsystems: graph input subsystem, wire frame subsystem, and polygon subsystem.

II. Kepler-Poinsot Solids

Four Kepler-Poinsot solids are listed in Table I-II, and illustrated in Fig. 1. The symbols $\{m, n\}$ in the tables stand for Schlölfli's symbols, which mean all the faces are congruent m-regular polygons (regular m-gons) and all the vertex figures are congruent n-regular polygons. Vertex figure of each vertex is the polygon formed by its adjacent vertices. It is similar to the segments joining the mid-point of the edges incident on the vertex, which is the conventional definition of vertex figure [1]. We define n-regular polygon by $n = 2\pi / \theta$ with its regularity, where θ is the exterior angle of each vertex. The exterior angle of pentagram is $4\pi / 5$, then pentagram can be regarded as 5/2-regular polygon.

Great dodecahedron {5, 5/2} and Great icosahedron {3, 5/2} are regular concave polyhedra with intersecting faces. They were introduced by L. Poinsot in 1809.

Great stellated dodecahedron {5/2, 3} and Small stellated dodecahedron {5/2, 5} are regular concave polyhedra with pentagram (5/2) as faces. They were introduced by J. Kepler in 1619. It depends on the definition of face of pentagram that their faces are intersecting or not. If the face of pentagram is defined by winding rule, faces are intersecting each other. On the other hand, if it is defined by even-odd rule, faces are not intersecting. Fig. 2 depicts the comparison of these two definitions.

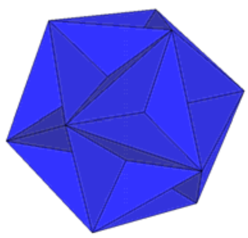
H. Nonaka is with Hokkaido University, Sapporo 060 0814, Japan (e-mail: nonaka@main.ist.hokudai.ac.jp).

TABLE I
THE LIST OF KEPLER-POINSOT SOLIDS (1)

Symbol	Polyhedron
{5,5/2}	Great Dodecahedron
{3,5/2}	Great Icosahedron
{5/2,3}	Great Stellated Dodecahedron
{5/2,5}	Small Stellated Dodecahedron

TABLE II
THE LIST OF KEPLER-POINSOT SOLIDS (2)

Symbol	Vertices	Edges	Faces
{5,5/2}	12	30	12
{3,5/2}	12	30	20
{5/2,3}	20	30	12
{5/2,5}	12	30	12


(a) Great dodecahedron {5,5/2}

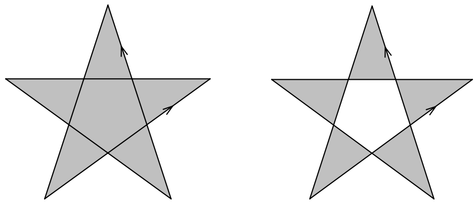

(b) Great icosahedron {3,5/2}


(c) Great stellated dodecahedron {5/2, 3}


(d) small stellated dodecahedron {5/2,5}

Triangles(3), pentagons(5), and pentagrams(5/2) are colored with yellow, blue, and gray, respectively.

Fig. 1 Four Kepler-Poinsot solids



(a) Winding rule (b) Even-odd rule

Fig. 2 Two alternative definitions of face or "inside" of pentagram

Traditionally, Kepler-Poinsot solids are formed and defined by stellating and extending of faces of dodecahedron or icosahedron as the core polyhedron. As is shown in Fig. 3, Small stellated dodecahedron, Great dodecahedron, and Great stellated dodecahedron are formed from Dodecahedron. By extending the faces of icosahedron, five faces meet again outside of Icosahedron and Great icosahedron is obtained. Fig. 4 shows the comparison of the size of the core icosahedron and Great icosahedron.

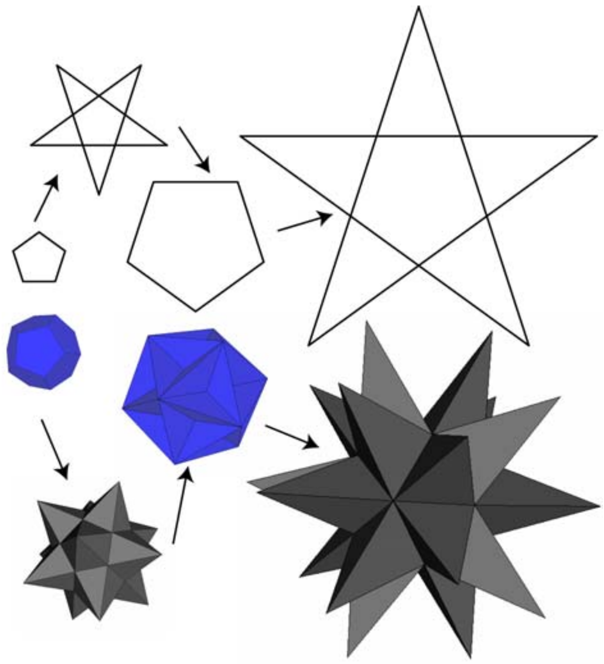


Fig. 3 Stellating and extending the face of dodecahedron

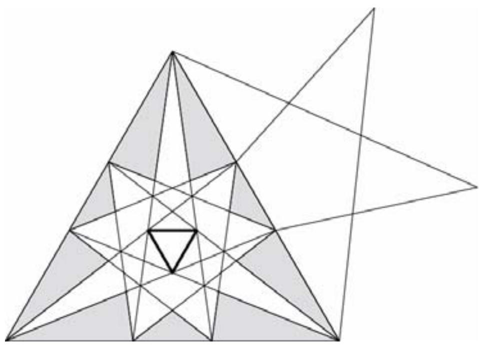
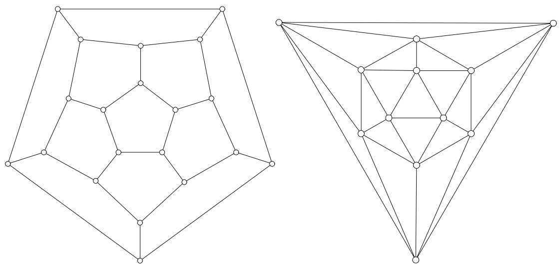


Fig. 4 Comparison of the size of the core icsahedron and Great icosahedron

III. Isomorphic Polyhedral Graph

Great dodecahedron fails to satisfy the Euler polyhedral equation. It has 12 pentagons, which is the origin of its name, but corresponding graph is isomorphic to icosahedron. Great icosahedron has 20 triangles, and the corresponding graph is also isomorphic to icosahedron. Great stellated dodecahedron has 12 pentagram, and the corresponding graph is isomorphic to dodecahedron. Small stellated dodecahedron has 12 pentagram, but the corresponding graph is isomorphic to icosahedron. It fails to satisfy the Euler polyhedral equation.

Fig. 5 illustrates the isomorphic graphs of Kepler-Poinsot solids. And Fig. 6 shows the screen shot of graph input subsystem.



(a) {5/2, 3} (b) {5, 5/2}, {3, 5/2}, {5/2, 5}

Fig. 5 Polyhedral graphs isomorphic to Kepler-Poinsot solids

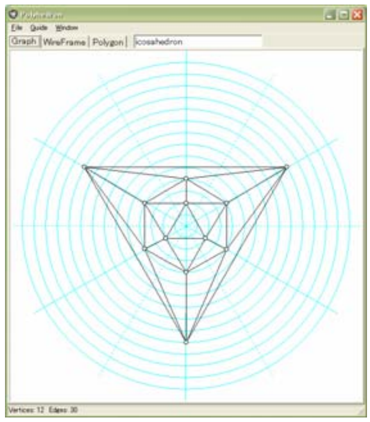
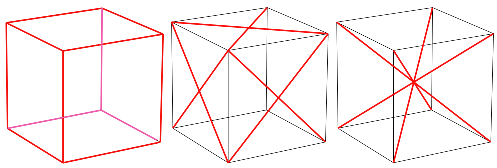


Fig. 6 Snapshot of graph input subsystem

IV. Wire Frame Polyhedra

In order to form a wire frame polyhedron from a corresponding graph, we define three binary relations between two vertices: adjacent, neighbor, and diameter (Fig. 7).



(a) adjacent (b) neighbor (c) diameter

Fig. 7 Binary relations between two vertices

$$adjacent(u,v) = \begin{cases} true & (u,v) \in E \\ false & otherwise \end{cases}$$

$$neighbor(u,v) = \begin{cases} true & (u,v) \notin E \wedge \\ & \quad \exists w.((u,w) \in E \wedge \\ & \qquad (v,w) \in E) \\ false & otherwise \end{cases}$$

$$diameter(u,v) = \begin{cases} true & length(u,v) = dmr(G) \\ false & otherwise \end{cases}$$

The relation `adjacent` corresponds to the equilaterality of edges. The relation `neighbor` corresponds to the vertex figure. The relation `diameter` corresponds to the circumsphere of polyhedron. Virtual springs are applied between vertices according to three relations and Hook's law. Let L and k be natural length of virtual spring and spring constant, and each subscript a, n, d corresponds to `adjacent`, `neighbor`, and `diameter`, respectively. $\boldsymbol{v}_i,\ i = 0,1,\cdots,p-1$ stands for the 3 dimensional coordinate of vertex $v_i \in V$. Then the total elastic potential E is given as follows.

$$E = E_a + E_n + E_d$$

$$E_a = \frac{k_a}{2} \sum_{i<j \wedge adjacent(v_i,v_j)} (L_a - \| v_i - v_j \|)^2$$

$$E_n = \frac{k_n}{2} \sum_{i<j \wedge neighbor(v_i,v_j)} (L_n - \| v_i - v_j \|)^2$$

$$E_d = \frac{k_d}{2} \sum_{i<j \wedge diameter(v_i,v_j)} (L_d - \| v_i - v_j \|)^2$$

Fig. 8 shows a screen shot of wire frame subsystem with tool windows, and Fig. 9 shows the wire frames of Kepler-Poinsot solids generated by the system using simulated elasticity.

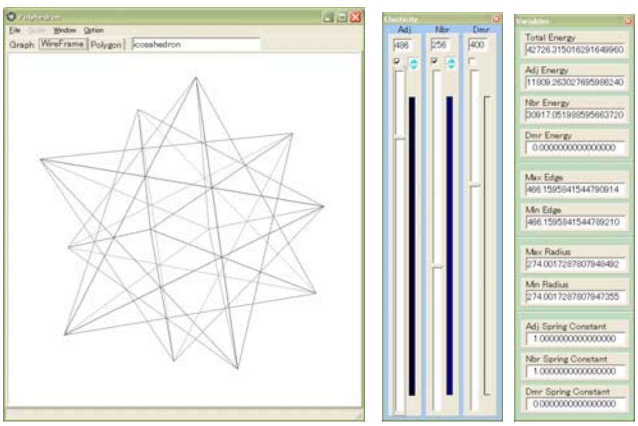
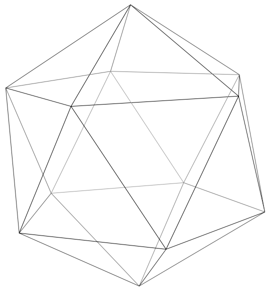
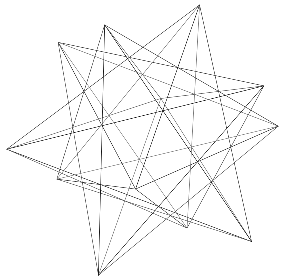


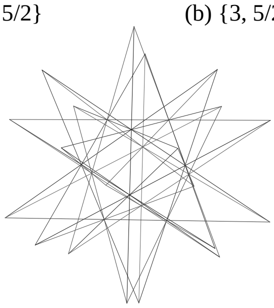
Fig. 8 Screen shot of wire frame subsystem with elasticity control tool and optional monitor window



(a) {5, 5/2}



(b) {3, 5/2}, {5/2, 5}



(c) {5/2, 3}

Fig. 8 Wire frames of Kepler-Poinsot solids

We can see in Fig. 8, wire frame of Great dodecahedron {5, 5/2} is identical to that of icosahedron {3, 5}. Wire frame of Great icosahedron {3, 5/2} and small stellated dodecahedron {5/2, 5} are also congruent. Vertex positions of above four polyhedral are equivalent. And vertex positions of Great stellated dodecahedron are equal to those of dodecahedron.

V. Modeling of Kepler-Poinsot solids

The final step is detecting and selecting appropriate faces. Fig. 9 shows snapshots of polygon subsystem.

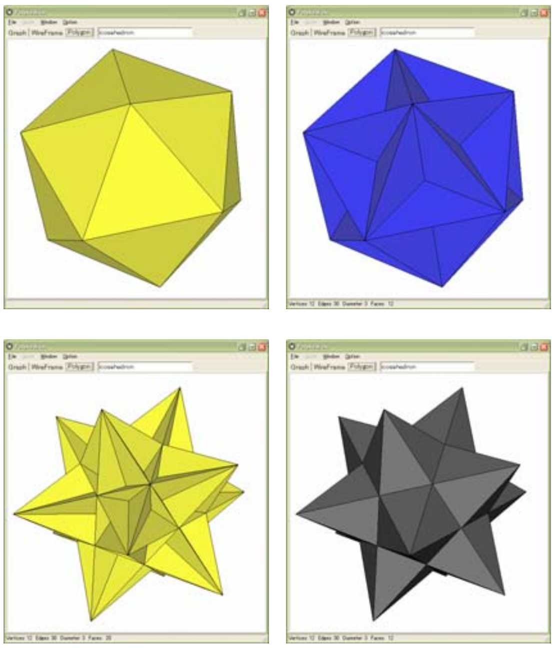


Fig. 9 Screenshots of polygon subsystem
Three Kepler-Poinsot solids generated from icosahedral graph

Detecting n-regular polygon is equivalent to finding simple closed path (cycle) with length n. Firstly, detecting triangles from Great icosahedron is equal to detecting triangles from icosahedron. Secondly, detecting pentagrams from Great stellated dodecahedron is common with detecting pentagons from dodecahedron. Lastly, detecting pentagons from Great dodecahedron is common with detecting pentagrams from Small stellated dodecahedron. In the last case, five vertices forming pentagon or pentagram are neighbor of another vertex, therefore, by reordering of these five vertices according to adjacency, correct face is detected.

The faces of Great dodecahedron and Great icosahedron are intersected each other, then hidden surface removal according the depth is required for rendering. *Z*-buffer algorithm is applicable for this purpose (for example [9]), but exterior elements of faces can be calculated by the coordinate of vertices, therefore, it can be achieved without *z*-buffer or graphics accelerator for real-time rendering.

In the case of Great dodecahedron, exterior surface is obtained as each pentagon subtracted by the pentagram formed by the same vertices. Exposed fragments of Great icosahedron are calculated as follows (Fig. 10).

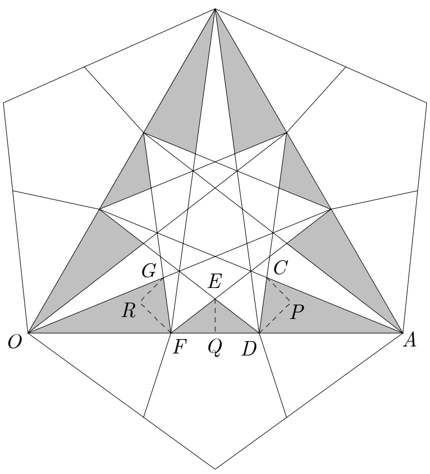


Fig. 10 Nine exposed fragments of triangle ΔOAB surrounding the great icosahedron (shadowed area)

Let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \cdots, \boldsymbol{r}$ be the position vector of each point $A, B, C, \cdots, R$ in Fig.10, then they are expressed by $\boldsymbol{a}$ and $\boldsymbol{b}$ as follows:

$$\boldsymbol{c} = \frac{5-\sqrt{5}}{5}\boldsymbol{a} + \frac{3\sqrt{5}-5}{10}\boldsymbol{b}, \quad \boldsymbol{d} = \frac{\sqrt{5}-1}{2}\boldsymbol{a}$$

$$\boldsymbol{e} = \frac{\sqrt{5}}{5}\boldsymbol{a} + \frac{5-2\sqrt{5}}{5}\boldsymbol{b}, \quad \boldsymbol{f} = \frac{3-\sqrt{5}}{2}\boldsymbol{a},$$

$$\boldsymbol{g} = \frac{5-\sqrt{5}}{10}\boldsymbol{a} + \frac{3\sqrt{5}-5}{10}\boldsymbol{b}.$$

$$\boldsymbol{p} = \frac{3+\sqrt{5}}{8}\boldsymbol{a} + \frac{3-\sqrt{5}}{8}\boldsymbol{b}, \quad \boldsymbol{q} = \frac{1}{2}\boldsymbol{a},$$

$$\boldsymbol{r} = \frac{1}{4}\boldsymbol{a} + \frac{3-\sqrt{5}}{8}\boldsymbol{b}$$

Three points P, Q, R are used for determining the depth order of three fragments, which form a concave region. ΔDPC and ΔFRG are right-angled triangles and axisymmetrical to ΔDQE and ΔFQE, respectively.

VI. Related Work

In order to form a wire frame polyhedron from a corresponding graph, simulated elasticity or virtual spring is used in several researches. For example, Chen et al. [10] applied imaginary springs between the vertices and the origin, and also between the pair of adjacent vertices. It is sufficient for triangular polyhedral or deltahedra. It include tetrahedron, octahedron, and icosahedron. Tyler [11] developed a tensetrity simulator “springie”, which can model arbitrary stable wire frame polyhedral by designing the structure to satisfy the condition of tensegrity. On the other hand, in the presented system, tensegrity is obtained semi-automatically using three binary relations between two vertices.

In the case of general polyhedra, the solution of face detection from an arbitrary planar graph is not unique. There are various researches for detecting polygons from wire frames, especially in the field of computer graphics. For example, Inoue et al. proposed an effective method of solid model reconstruction from planar graphs with exhaustive collection and pruning down [12]. On the other hand, in our purpose, the graphs are limited to icosahedral and dodecahedral graphs, therefore, the algorithm is quite simple. The same approach to semi-regular polyhedra has been presented by the author [13].

There is also theoretical research on the relation between polyhedral graph and abstract polyhedra by Grünbaum [14].

References

[1] H. S. M. Coxeter, “Regular Polytopes”, Dover Publications, 1973
[2] P. Martin, and D. Martin, “PolyFormes: Software for the declarative modelling of polyhedra”, The Visual Computer, 15 (2), pp. 55-76, 1999
[3] G. W. Hart, and H. Picciotto, “Zome Geometry: Hands on Learning With Zome Models”, Key Curriculum Pr., 2001.
[4] G. W. Hart, “Virtual Polyhedra --- The Encyclopedia of Polyhedra”, http://www.georgehart.com/virtual-polyhedra/vp.html
[5] G. W. Hart, “Conway Notation for Polyhedra”, available at the URL: http://www.georgehart.co/virtual-polyhedra/conway_notation.html”
[6] A. Holden, “Shapes, Space, and Symmetry”, Columbia University Press, 1971
[7] H. S. M. Coxeter, et al., “Fifty-Nine Icosahedra”, Tarquin Publications, Norfolk, England, 1999
[8] J. Skilling, “The Complete Set of Uniform Polyhedra”, Philosophical Transactions of the Royal Society of London, Series A, Mathematical and Physical Science, 278, pp. 111-135, 1975
[9] T. A. Möller, and E. Haines, “Real-time Rendering”,A K Peters, Ldt., 2002
[10] J. X. Chen, Y. Yang, and X. Wang, “Phisics-Based Modeling and Real-Time Simulation”, Computing in Science & Engineering, 3, pp. 98-102, 2001.
[11] T. Tyler, “Springie: A tensegrity simulator using Java, VRML and POV Ray”, http://springie.com/
[12] K. Inoue, K. Shimada, and K. Chilaka, “Solid Model Reconstruction of Wireframe CAD Models Based on Topological Embeddings of Planar Graphs”, Journal of Mechanical Design, **125**, pp. 434-442, 2003.
[13] H. Nonaka, “Detection of Faces in Wire-Frame Polyhedra”, Proceedings of International Conference on Computer Graphics Theory and Applications, pp. 134-137, Jan. 2008.
[14] B. Grünbaum, “Graphs of polyhedra; polyhedra as graphs”, Discrete Mathematics, **307**, pp. 445-463, 2007.