Animation Visualization for Vertex Coloring of Polyhedral Graphs

Hidetoshi Nonaka
Graduate School of Information Science and Technology, Hokkaido University
Sapporo, 060 0814, Japan
nonaka@main.ist.hokudai.ac.jp

ABSTRACT

Vertex coloring of a graph is the assignment of labels to the vertices of the graph so that adjacent vertices have different labels. In the case of polyhedral graphs, the chromatic number is 2, 3, or 4. Edge coloring problem and face coloring problem can be converted to vertex coloring problem for appropriate polyhedral graphs.

We have been developed an interactive learning system of polyhedra, based on graph operations and simulated elasticity potential method, mainly for educational purpose.

In this paper, we introduce a learning subsystem of vertex coloring, edge coloring and face coloring, based on minimum spanning tree and degenerated polyhedron, which is introduced in this paper.

Keywords: Vertex Coloring, Polyhedral Graph, Animation, Visualization, Interactivity

1. INTRODUCTION

Vertex coloring of a graph is the assignment of labels to the vertices of the graph so that adjacent vertices have different labels [1-3]. The 4-colour theorem proved by Appel and Haken in 1977, indicates that every planar graph is 4-colourable. Every polyhedral graph is 3-connected planar graph, according to the theorem by Steinitz. Therefore, it is also 4-colourable. Consequently, the chromatic number of a polyhedral graph is 2, 3, or 4. There are various coloring methods, for example, greedy coloring algorithm, sequential coloring algorithm, distributed algorithm, decentralized algorithm, and so on. Determination of 2-colourability is equivalent to testing bipartiteness, therefore, it is computable in linear time. However, in the case of more than 2 coloring, the computational complexity is known to be NP-complete, even for 3-colourability [4], and 4-colourability [5].

The author has been developed an interactive learning system of polyhedra, based on graph operations and simulated elasticity potential method, mainly for educational purpose [6-10]. By using this system, the user or the learner can make and handle various polyhedra, including Platonic solids, Archimedean solids [9], Kepler-Poinsot solids [7], fullerenes molecular structures, and geodesic dome constructions.

In this paper, we introduce a learning subsystem of interactive vertex coloring, edge coloring, and face coloring, based on minimum spanning tree and degenerated polyhedron. Vertex coloring of polyhedral graph itself is trivial in a mathematical sense, and it is not novel also in a practical sense. However, visibility and interactivity can be helpful for the user to understand intuitively the mathematical structure and the computational scheme, by visualizing the process of the calculation, and by allowing the user to contribute the computation.

2. POLYHEDRON MODELING SYTEM

In this section, we summarize the system of interactive modeling of polyhedra described in [6-10]. It consists of three subsystems: graph input subsystem, wire-frame subsystem, and polygon subsystem.

2.1 Graph Input Subsystem

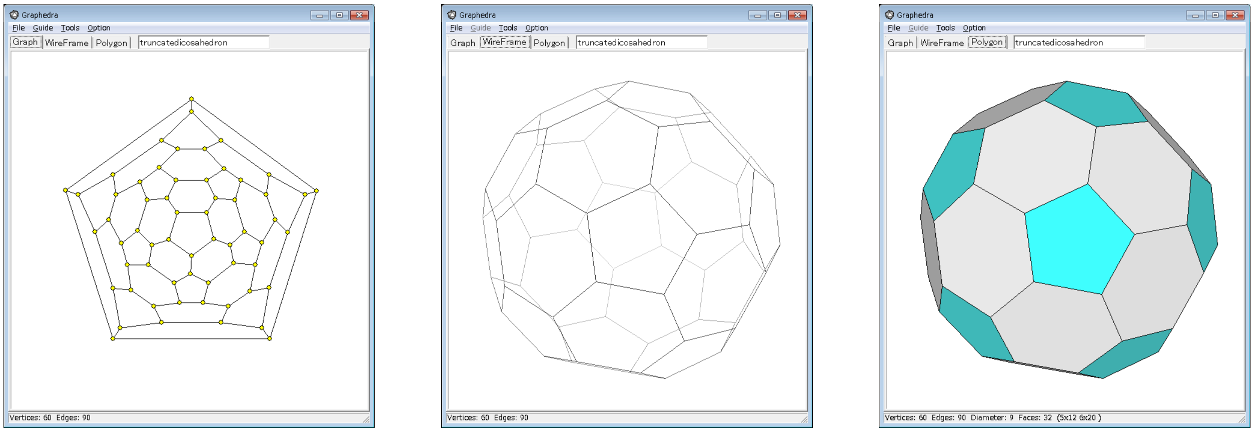
Figure 1(a) shows a screen shot of graph input subsystem, where a graph isomorphic to truncated icosahedrons is drawn. The first step of the modeling of polyhedron is drawing a polyhedral graph isomorphic to the intended polyhedron. In the subsystem, vertex addition, vertex deletion, edge addition, and edge deletion are implemented as fundamental operations. Some additional utilities are also implemented such as grid lines, grid snapping, vertex coloring according to degrees, and so on.

2.2 Wire-Frame Subsystem

Figure 1(b) shows a screen shot of wire-frame subsystem. After constructing a polyhedral graph the next step is arranging vertices in 3D space with virtual springs and Hooke's law. Wire-frame polyhedron can be formed by controlling the natural length of virtual spring corresponding to three types of binary relations between pairs of vertices.

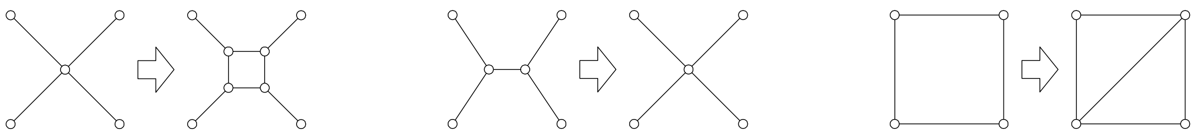
2.3 Polygon Subsystem

Figure 1(c) shows a screen shot of polygon subsystem. After arranging vertices in 3D space, the last step is detecting faces, selecting appropriate faces, and rendering the solid. Detecting n-polygon is equivalent to finding simple closed path with length n. Some additional utilities such as opening faces, meshed faces are implemented.



(a) (b) (c)

Figure 1. Screen shots of Interactive Polyhedron Modeling System.
(a) Graph input subsystem, (b) Wire-frame subsystem, and (c) Polygon subsystem.



(a) (b) (c)

Figure 2. Three Graph Operations.
(a) Vertex splitting, (b) Edge contraction, and (c) Diagonal addition.

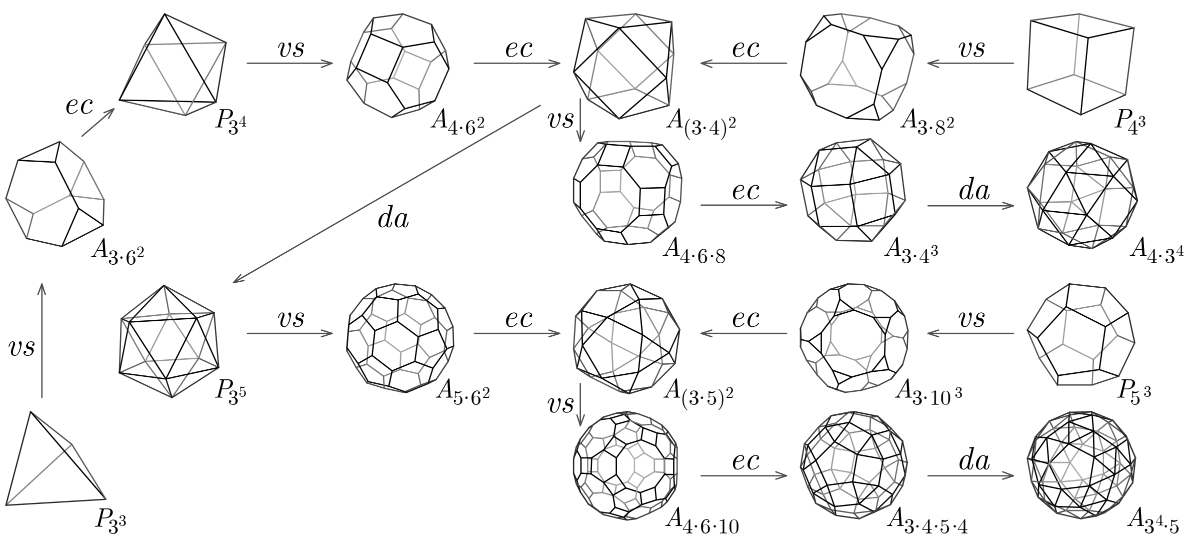


Figure 3. Relations of 5 Platonic graphs and 13 Archimedean graphs using 3 operations.

2.4 Graph Operation for Polyhedral Graph

Three graph operations are defined for polyhedral graphs: *vertex splitting*, *edge contraction*, and *diagonal addition* (Figure 2) [9]. By these three operations, 5 regular polyhedra (Platonic solids) and 13 semi-regular polyhedra (Archimedean solids) are interconnected as shown in Figure 3. By using these operations, the user can model various polyhedra from one seed polyhedron.

3. ANIMATION VISUALIZATION OF VERTEX COLORING

Table 1 in the next page is the complete list of regular polyhedra and semi-regular polyhedra. Symbols *vc*, *ec*, and *fc* are the chromatic numbers of vertex coloring, edge coloring and face coloring. Face coloring of a planar graph *G* is equivalent to vertex coloring of the dual of *G*. Edge coloring of a polyhedral graph *G* is equivalent to vertex coloring of the *ambo* of *G*. Ambo is one of Conway Polyhedron notations [10].

A graph *G* is *k*-colorable if and only if *G* is *k*-partite. In the case of polyhedral graph, *k* can be 2, 3 or 4. By identifying vertices in each part of *k*-partite graph, one of line segment (1-simplex), triangle (2-simplex), or tetrahedron (3-simplex) is obtained. In this paper, we call such a polytope generated from a polyhedron, *degenerated* polyhedron. Figure 4 shows three examples of degenerated polyhedra: snubdodecahedron (to tetrahedron), small rhombicosidodecahedron (to triangle), and great rhombicosidodecahedron (to line segment).

The process of interactive vertex coloring is as follows. When the user presses the "Degenerate" button, the system tests the chromatic number in the order of 2, 3, and 4 colorabilities. If the graph is *k*-colorable, it is partitioned to *k*-partite graph, and degenerated to one of line segment, triangle, or tetrahedron, with animations (Figure 5 (a-d)). After the user select different color for each vertex of degenerated polyhedron (Figure 5 (e)), when the user presses the "Release" button, the polyhedron recovers the original shape with also animations (Figure 5 (f-h)). The user or learner can observe how the colors are assigned to the vertices so that adjacent vertices have different colors, and also understand unconsciously that *k*-colorable and *k*-partite are equivalent.

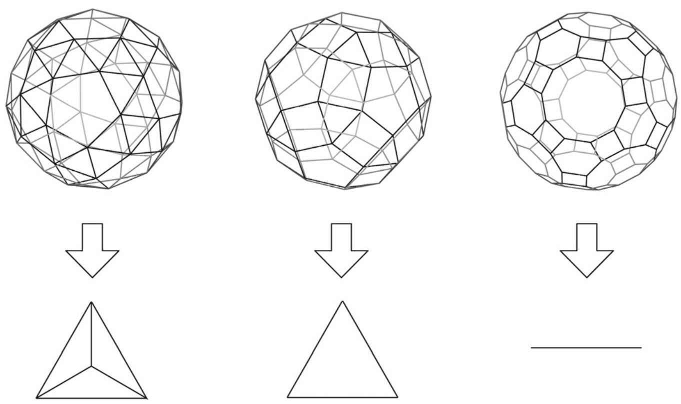


Figure 4. Examples of degenerated polyhedra (polytopes).

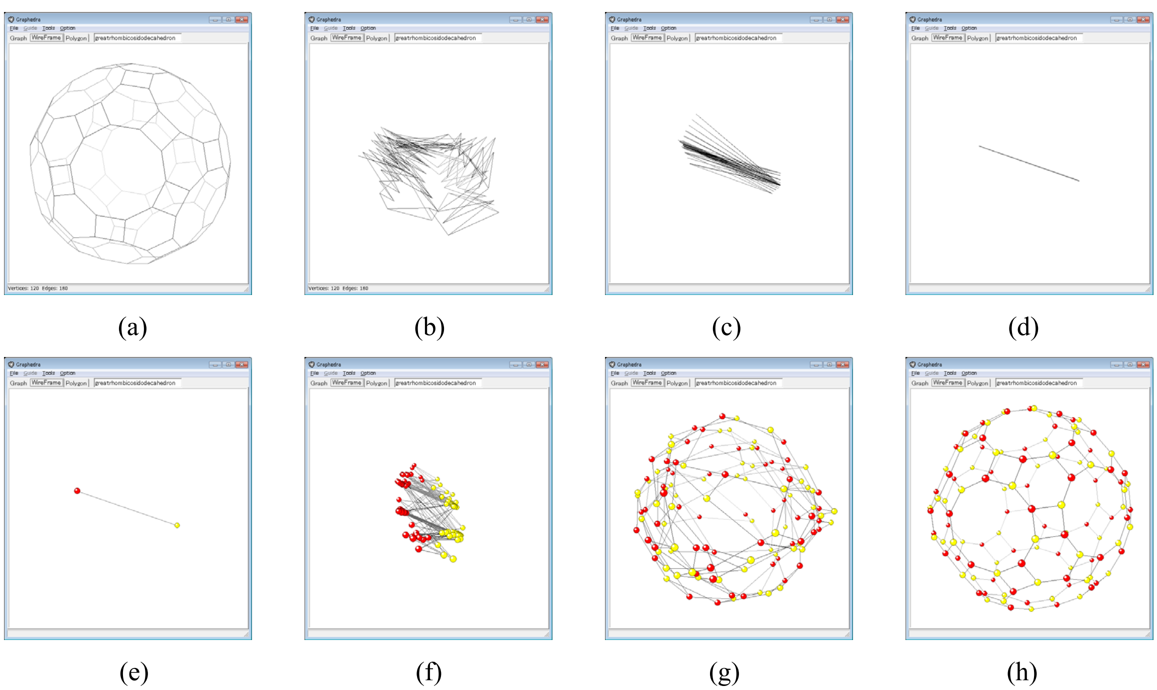


Figure 5. An example of interactive vertex coloring (great rhombicosidodecahedron: line segment).

4. CONCLUSION

In this paper, interactive vertex coloring system for polyhedral graph has been presented. It was developed as a subsystem of an interactive learning system of polyhedra, based on graph theory. The learner can not only observe how the colors are assigned to the vertices, but also understand unconsciously that the notions of *k*-colorable and *k*-partite are equivalent.

REFERENCES

[1] Bondy, J.A. and Murty, U.S.R. (2008) Graph Theory, Springer.

[2] Gross, J. and Yellen, J. (1998) Graph Theory and Its Applications, CRC Press.

[3] Nishizeki, T. and Chiba, N. (1988) Planar Graphs, Dover Publications Inc.

[4] Beigel, R. (2005) 3-Coloring in time $O(1.3289^n)$, *Journal of Algorithms* **54**(2), 168-204.

[5] Byskov, J.M. (2004) Enumerating maximal independent sets with applications to graph colouring, *Operations Research Letters*, **32**, 547-556.

[6] Nonaka, H. (2008) Detection of Faces in Wire-Frame Polyhedra, *Proceedings of International Conference on Computer Graphics Theory and Applications*, **3**, 134-137.

[7] Nonaka, H. (2008) Modeling of Kepler-Poinsot Solid Using Isomorphic Polyhedral Graph, *International Journal of Computer Science*, **3**(1), 64-67.

[8] Nonaka, H. (2008) Interactive Modeling of Uniform Polyhedra by Means of Graph Drawing, *Proceedings of International Conference on Information Technology and Applications*, **5**, 822-827.

[9] Nonaka, H. (2008) Operation-Based Notation for Archimedean Graph, *Journal of Systemics, Cybernetics and Informatics*, **6**(6), 19-22, 2008.

[10] Nonaka, H. (2009) Visualization of Conway Polyhedron Notation, *Proceedings of World Congress on Science, Engineering and Technology*, **38**, 692-695.

Table 1. The list of regular polyhedra (Platonic solids) and semi-regular polyhedra (Archimedean solids).
v, e, and f stand for the numbers of vertices, edges and faces, respectively.
vc, ec, and fc are the chromatic numbers of vertex-coloring, edge-coloring and face-coloring.

Symbol	Name of polyhedron	v	e	f	vc	ec	fc
P_{3^3}	Tetrahedron	4	6	4	4	3	4
P_{4^3}	Cube	8	12	6	2	3	3
P_{3^4}	Octahedron	6	12	8	3	3	2
P_{5^3}	Dodecahedron	20	30	12	3	3	4
P_{3^5}	Icosahedron	12	30	20	4	3	3
$A_{(3\cdot4)^2}$	Cuboctahedron	12	24	14	3	3	2
$A_{4\cdot6\cdot10}$	Great Rhombicosidodecahedron	120	180	62	2	3	3
$A_{4\cdot6\cdot8}$	Great Rhombicuboctahedron	48	72	26	2	3	3
$A_{(3\cdot5)^2}$	Icosidodecahedron	30	60	32	3	3	2
$A_{3\cdot4\cdot5\cdot4}$	Small Rhombicosidodecahedron	60	120	62	3	3	2
$A_{3\cdot4^3}$	Small Rhombicuboctahedron	24	48	26	3	3	2
$A_{3^4\cdot4}$	Snub Cube	24	60	38	3	3	3
$A_{3^4\cdot5}$	Snub Dodecahedron	60	150	92	4	3	3
$A_{3\cdot8^2}$	Truncated Cube	24	36	14	3	3	4
$A_{3\cdot10^2}$	Truncated Dodecahedron	60	90	32	3	3	4
$A_{5\cdot6^2}$	Truncated Icosahedron	60	90	32	3	3	4
$A_{4\cdot6^2}$	Truncated Octahedron	24	36	14	2	3	3
$A_{3\cdot6^2}$	Truncated Tetrahedron	12	18	8	3	3	4