A New Paradigm for Changing Topology of 2-Manifold Polygonal Meshes

ERGUN AKLEMAN *
Visualization Laboratory
College of Architecture

JIANER CHEN †
Department of Computer Science
College of Engineering

VINOD SRINIVASAN ‡
Visualization Laboratory
College of Architecture

Texas A&M University

Abstract

In this paper, we present a new paradigm that allows dynamically changing the topology of 2-manifold polygonal meshes. Our new paradigm always guarantees topological consistency of polygonal meshes. Based on our paradigm, by simply adding and deleting edges, handles can be created and deleted, holes can be opened or closed, polygonal meshes can be connected or disconnected.

These edge insertion and edge deletion operations are highly consistent with subdivision algorithms. In particular, these operations can easily be included into a subdivision modeling system such that the topological changes and subdivision operations can be performed alternatively during model construction.

We demonstrate practical examples of topology changes based on this new paradigm and show that the new paradigm is convenient, effective, efficient, and friendly to subdivision surfaces.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, Surface, Solid and Object Representations; J.6 [Computer Applications]: Computer Aided Engineering—Computer Aided Design (CAD)

Keywords: Shape Modeling, Topological Modeling, Subdivision Surfaces, Implicit Surfaces

1 Introduction

One of the important problems in modeling is to be able to change an objects's topology during construction. Currently, this goal can only be achieved by implicit related methods [31, 32, 29, 38, 22, 4, 19, 14]. However, the computation of shapes represented implicitly is extremely expensive. Parametric representations have long been popular for modeling mainly because they allow real time shape computation [3]. However, they do not permit topological change during construction. Moreover, widely used tensor product parametric representations do not even provide a large variety of topologies [27] since control meshes of tensor product parametric surfaces must be organized as a regular rectangular structure.

Subdivision methods [5, 11, 26, 20, 34, 33, 10] are becoming popular for modeling smooth surfaces since they solve the fundamental problem of tensor product parametric surfaces without sacrificing the speed of shape computation. Unlike tensor product surfaces, in subdivision surfaces, the control meshes do not have to have a regular rectangular structure. Subdvision algorithms can smooth out 2-manifold (or 2-manifold with boundary) polygonal meshes [39, 16].

Although the subdivision paradigm has been around more than two decades [5, 11, 26], subdivision methods have not been popular until recently. One of the chief arguments against subdivision methods was that unlike parametric surfaces they do not admit analytical form [28]. This argument turned out to be incorrect since Stam [33] has recently proved that Catmull-Clark and Loop subdivision schemes can be directly evaluated without explicit subdivision. Despite their power, subdivision methods have two main restrictions.

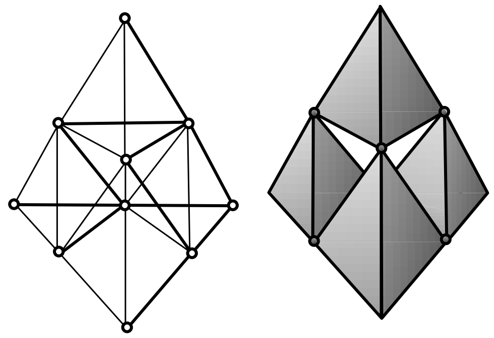

Figure 1: A non-manifold polygonal mesh described by the union of four tetrahedra with shared vertices.

- Subdivision methods can accept only control meshes that are either 2-manifold or 2-manifold with boundary [39]. Non-manifold control meshes or disjoint control meshes that represent a single object are not accepted by subdivision algorithms. An example of non-manifold control meshes is shown in Figure 1. This union of four tetrahedra is non-manifold because of the vertices shared by tetrahedra. Figure 2 shows an example of disjoint control meshes that represent a single object, the set difference of two rectangular prisms. In solid modeling, there exists various internal data representations to support such non-manifold shapes [23].

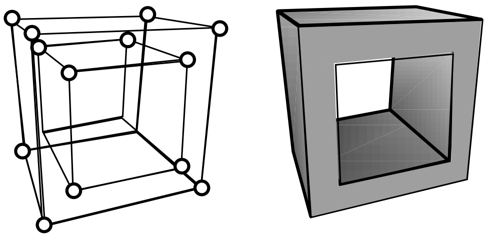

Figure 2: Two disjoint meshes that represent a single object described by the set difference of two rectangular prisms.

*Address: 216 Langford Center, College Station, Texas 77843-3137. email: ergun@viz.tamu.edu. Supported in part by the Texas A&M, Scholarly & Creative Activities Program.

†Address: Department of Computer Science, College Station, TX 77843-3112. email: chen@cs.tamu.edu. Supported in part by the National Science Foundation under Grant CCR-9613805.

‡Address: 216 Langford Center, College Station, Texas 77843-3137. email: vinod@viz.tamu.edu. Supported in part by the Texas A&M, Interdisciplineary Activities Program.

- During subdivision modeling it is not possible to change the topology of a shape. One possible solution to change the topology is to use an implicit based approach. For instance, if the initial meshes are defined implicitly by set operations, new polygonal meshes that are topologically different can be obtained by set operations. However, this approach is not really a solution for changing topology since the set operations over 2-manifold surfaces can result in the kind of non-manifold shapes or disconnected meshes shown in Figures 1 and 2, and these shapes cannot be smoothed by subdivision algorithms. Thus, there is a need for a new and completely different paradigm.

In this paper, we present a purely "polygonal" and "non-implicit" approach changing the topology of polygonal meshes. The topological changes we demonstrate include opening and closing holes, creating and deleting handles, connecting two disjoint meshes into one, and separating one mesh into two disconnected ones. These topological changes are simply done by inserting or deleting edges to a polygonal mesh. During these operations, we also guarantee that polygonal meshes remain valid 2-manifolds. Therefore, unlike implicitly based topological operations, our operations are subdivision friendly, i.e., subdivision operations can always smooth out the meshes obtained by our operations.

Our approach to handling topological changes makes a natural partner with the subdivision approach within a 3D modeling system. Our approach is not only suitable for modeling initial 2-manifold polygonal meshes, but also useful for changing the topology of a 2-manifold polygonal mesh that are smoothed by subdivision operations. Since subdivision operations are essential for the improvement of the quality of meshes, performing the topological change operations and the subdivision operations alternatively provides a powerful shape modeling approach that supports a hierarchy of topology changes and quality improvements at different levels of details.

Our approach is based on graph rotation systems. A graph with a rotation system is embedded to a unique 2-manifold shape [12]. Therefore, it always guarantees the representation of valid 2-manifold polygonal meshes. Moreover, it is easy to change the topology of meshes by simply inserting or deleting edges in the corresponding graph rotation system. An edge insert operation can either (1) combine two faces by inserting a hole, (2) combine two faces by adding a handle, (3) combine two faces by joining two separate 2-manifold meshes, or (4) subdivide a single face into two. Conversely, an edge delete operation can either (1) separate one face into two by closing a hole, (2) separate one face into two by deleting a handle, (3) separate one face into two by disconnecting one 2-manifold object into two, or (4) combine two faces into one. In either operation the first three cases change the topology of a mesh. The last one does not change the topology; it only increases the number of polygons.

We first introduce the previous works in graph rotation system. Then, we demonstrate why and how edge insert and edge delete operations work by illustrating each of these cases with figures. The rest of the paper covers implementation issues and examples.

2 Graph Rotation Systems

With the introduction of subdivision surfaces and wider usage of implicit surfaces, topology has become an important element of computer graphics research, development and production. There has been various studies in topological modeling during the last decade [15, 36, 35, 17].

Our techniques introduced in this paper is based on *graph rotation systems*, which is new in the study of topological modeling. The graph rotation system is a powerful tool for guaranteeing topological consistency and provides great convenience in topology changes. In this section, we introduce historical background and some mathematical fundamentals for graph rotation systems (see [18] for more detailed discussion).

The concept of rotation systems of a graph originated from the study of graph embeddings and it is implicitly due to Heffter [21] who used it in Poincare dual form. A graph embedding in an orientable surface corresponds to an obvious rotation system, namely, the one in which the rotation at each vertex is consistent with the cyclic order of the neighboring vertices in the embedding. Edmonds [12] was the first to call attention explicitly to studying rotation systems of a graph.

Let G be a graph. A *rotation* at a vertex v of G is a cyclic permutation of the edge-ends incident on v. A *rotation system* of G is a list of rotations, one for each vertex of G. Given a rotation system of a graph G, to each oriented edge (u, v) in G one assigns the oriented edge (v, w) such that vertex w is the immediate successor of vertex u in the rotation at vertex v. The result is a permutation on the set of oriented edges, that is, on the set in which each undirected edge appears twice, once with each possible direction. In each edge-orbit under this permutation, the consecutive oriented edges line up head to tail, from which it follows that they form a directed cycle in the graph. If there are r oriented edges in an orbit, then an r-sided polygon can be fitted into it. Fitting a polygon to every such edge-orbit results in polygons on both sides of each edge, and collectively the polygons form a 2-manifold.

Edmonds [12] has shown that every rotation system of a graph gives a *unique* orientable 2-manifold. Moreover, the corresponding orientable 2-manifold is constructible, as we described above. For any 2-manifold S, there is a graph G and a rotation system ρ of G such that ρ corresponds to an embedding of G on S [6].

Therefore, the existence of the bijective correspondence between graph embeddings on orientable 2-manifolds and graph rotation systems enables us to represent topological objects by combinatorial ones. In particular, every 2-manifold can be represented by a rotation system of a graph, and every rotation system of a graph corresponds to a valid 2-manifold. In consequence, the presentation of graph rotation systems always guarantees topological consistency. Recently, we have developed a very efficient algorithm that, given a rotation system of a graph, constructs the corresponding 2-manifold [7].

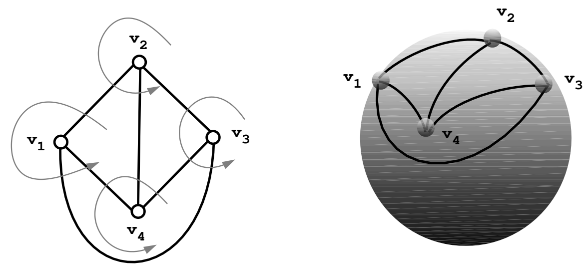

Figure 3: A graph drawn in a rotation system and the corresponding embedding on the sphere.

Based on graph rotation systems, a graph that is drawn in 2D can uniquely represent a 2-manifold shape. For example, consider two graphs given in Figures 3 and 4. These graphs are drawn in such a way that the rotation at each vertex can be traced by traversing the incident edges in counter-clockwise order. These are two different rotation systems for the same graph. When we consider the rotation order they correspond to two different 2-manifold shapes: the sphere in Figure 3 and the toroid in Figure 4.

The graph rotation system also provides a concise and simple internal data representation as a set of vertices and a rotation order

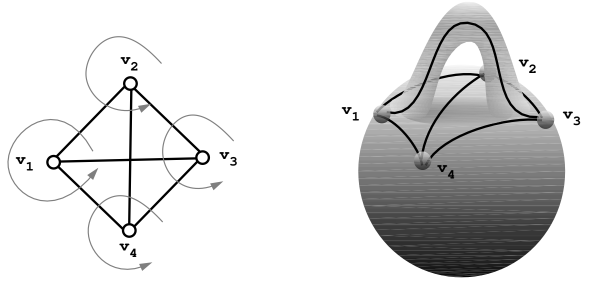

Figure 4: The same graph as in Figure 3 drawn in a different rotation system and the corresponding embedding on the toroid.

for each vertex. For example, the rotation system of the graph in Figure 3 is given by the ordered lists:

$$v_1: \; v_4v_2v_3, \qquad v_2: \; v_3v_1v_4,$$
$$v_3: \; v_1v_2v_4, \qquad v_4: \; v_1v_3v_2,$$

and the rotation system of the graph in Figure 4 is given by ordered lists:

$$v_1: \; v_4v_3v_2, \qquad v_2: \; v_3v_1v_4,$$
$$v_3: \; v_4v_2v_1, \qquad v_4: \; v_1v_3v_2.$$

Rotation systems can easily be implemented as a set of linked lists. We should also point out that each edge of a graph appears exactly twice in any of its rotation systems. Therefore, the amount of computer memory used for this representation is small.

Based on a rotation system, each face can easily be constructed by traversing the face boundary, following the rotation order of the vertices on the boundary, as shown in Figure 5. By a *face-corner* of a face, we refer to a vertex on the face boundary, plus the two neighboring edges.

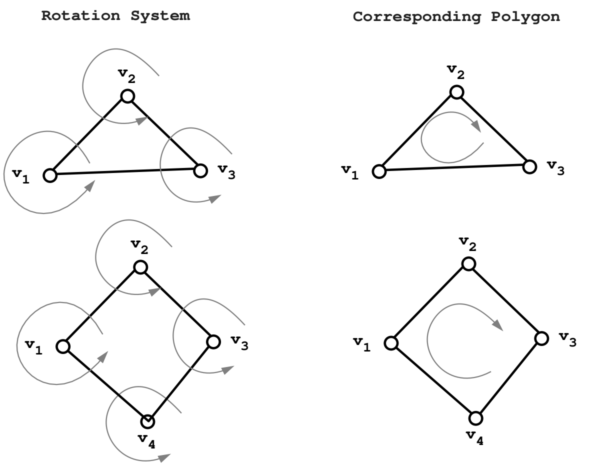

Figure 5: Rotation order of faces is provided by rotation system.

3 Inserting or Deleting an Edge in a Graph Rotation System

Any rotation order of two vertices of a graph rotation system can be given by the following representation without loss of generality:

$$v_1: \; v_3v_2\alpha, \qquad v_4: \; v_6v_5\beta$$

where α and β are any sequence of vertices. Let an edge be inserted between v_1 and v_4 as shown in Figure 6. The new graph rotation system can easily be obtained by simply updating the rotation orders of v_1 and v_4 as follows:

$$v_1: \; v_3v_4v_2\alpha, \qquad v_4: \; v_6v_1v_5\beta$$

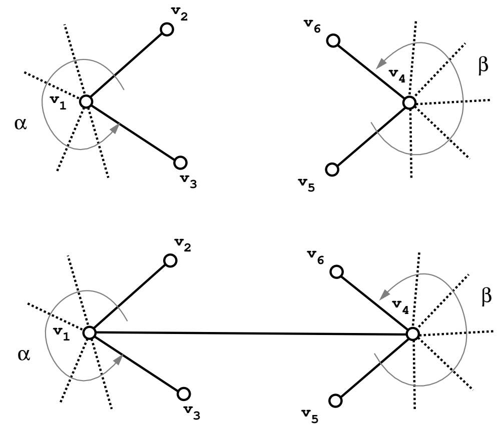

Figure 6: Inserting an edge.

Although this insert operation is extremely simple, it changes the faces of the 2-manifold object represented by the graph rotation system. It either connects two faces by "merging" them into one single face or subdivides a single face into two faces. These two cases are illustrated in more detail as follows:

- The face-corners $\langle v_3, v_1, v_2\rangle$ and $\langle v_6, v_4, v_5\rangle$ do not belong to the same face. In this case, the insert operation connects the two faces by merging them into one single face as shown in Figure 7.
- The face-corners $\langle v_3, v_1, v_2\rangle$ and $\langle v_6, v_4, v_5\rangle$ belong to the same face. In this case, the insert operation subdivides the face into two as shown in Figure 8.

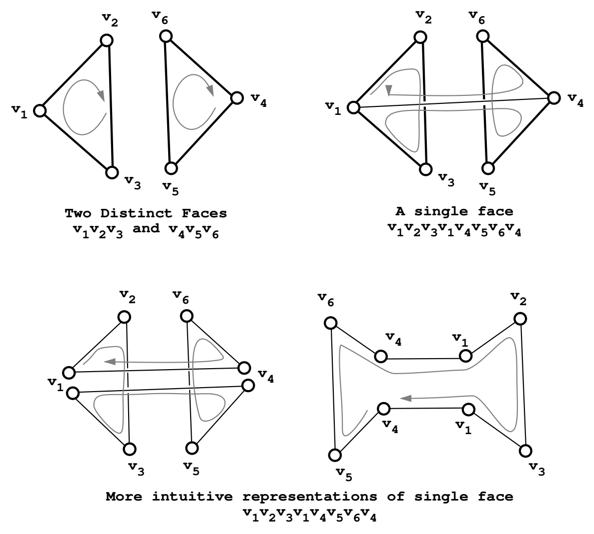

Figure 7: Inserting an edge between two different faces merges the two faces.

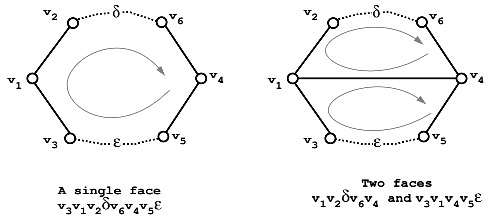

Figure 8: Inserting an edge between two vertices of the same face divides it into two faces.

The edge deletion operation can be elaborated in a similar manner. Basically, if an edge has its two sides in different faces, then deleting the edge merges the two faces into a larger one. On the other hand, if the two sides of the edge are in the same face, then deleting the edge separates the face into two faces.

4 Changing Topology with Graph Rotation Systems

Let there be two faces that are described by a rotation system. We assume that these two faces are not the same. However, they can share a vertex, an edge or several edges. They may also belong to two disconnected 2-manifolds. Based on the rotation system, any edge inserted by connecting these two faces converts them into one single face as we have shown in Figure 7.

If these two faces belong to two disconnected 2-manifold shapes, after the insert edge operation these two 2-manifold shapes become connected by the new inserted edge. If the faces belong to the same 2-manifold, the edge insert operation increases the genus of the 2-manifold by one, i.e., a handle (or a hole) is added.

In both cases, a new face created by an edge insert operation is, in fact, a handle as shown in Figure 9. Therefore, we call this type of face *"face-handle"*. It is not easy to visualize this face-handle, however the structure of the handle becomes apparent when the face is smoothed by subdivision operations as represented in Figure 9.

If this handle goes through the inside of a 2-manifold, it becomes a hole. If the handle goes outside of the 2-manifold, it becomes a handle. The hole and the handle are automatically created or deleted based on geometry and edge insertion/deletion rules we described in previous section.

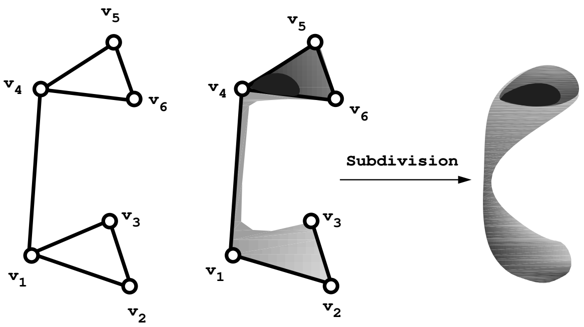

Figure 9: Handle becomes apparent after subdivision operations.

The handle that results from a subdivision operation includes an extraordinary vertex with a high valance. (The term valance is used to denote the number of edges incident on a vertex. Valance of an extraordinary vertex on a handle is at least equal to 7). In most cases, the resulting subdivision surface will not be C^1 continuous at this extraordinary vertex because of the unusual structure of our face-handles.

Fortunately, it is possible to improve the quality of the handle simply by inserting new edges. As shown in Figure 10 if a second edge is inserted to connect any two vertices of the face-handle, the new edge separates the face-handle into two faces without changing the topology of the mesh. On the other hand, the quality of the handle is usually improved.

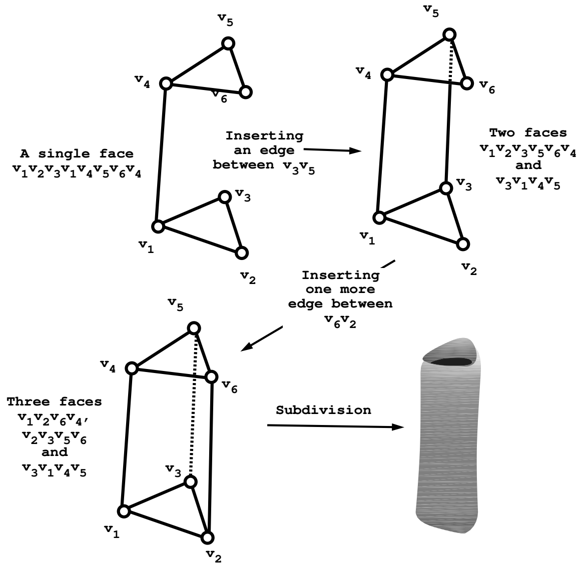

Figure 10: Improving the quality of handles by inserting more than one edge.

5 Implementation

As we have stated in Section 2, rotation systems can easily be implemented as a set of linked lists. This structure is extremely simple and it can easily be implemented. The only practical problem with using such a structure in an interactive system is that it requires construction of faces in each rendering step.

Rotation systems can also be implemented by using a Doubly Connected Edge List (DCEL), which is widely used in computation geometry [30]. Edge insertion and deletion operations under DCEL take linear time in the worst case [8].

Instead, in our implementation, we use a Doubly Linked Face List (DLFL), a data structure we have proposed theoretically earlier [1]. A DLFL structure always corresponds to a graph rotation system. Therefore, it always guarantees valid topological consistency [1]. The main reason that we use DLFL is that it supports logarithmic time edge insertion and deletion operations "with face construction". With the DLFL data structure, subdivision operations can also be implemented easily.

To show the feasibility of our approach, we have created various 3D models that show the connection of several 2-manifolds and the creation of handles and holes. In all our examples, we first convert irregular meshes into meshes that consist of only quadrilaterals to obtain initial mesh. We, then, smooth out this initial mesh by using Catmull-Clark subdivision scheme [5].

- Figure 11 shows the connection of four disconnected tetrahedra. All faces of the tetrahedra are first subdivided to make each face a quadrilateral. Then, the inside faces of every two neighboring tetrahedra are connected by an infinitely small

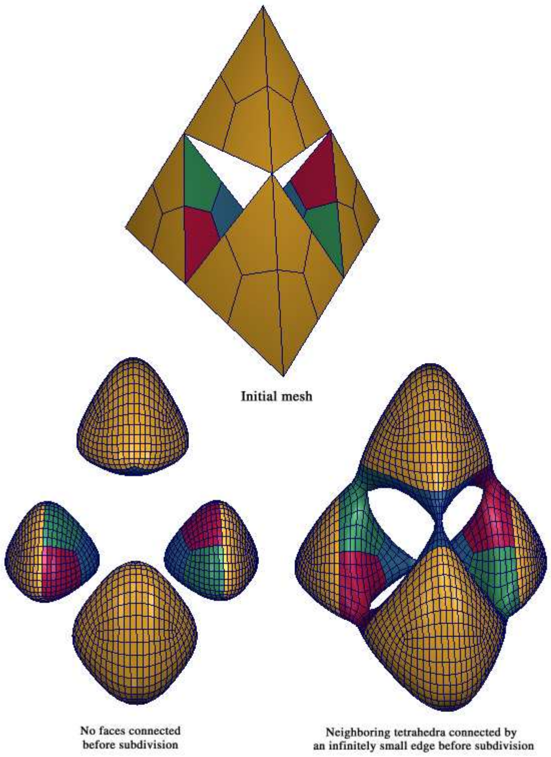

Figure 11: Connecting four tetrahedra.

edge. It is interesting to note that these 3D shapes may "look" non-manifold. They are actually 3D representations of a rotation system which is a representation of a 2-manifold. Therefore, we can apply subdivision algorithms. On the other hand, if they had been represented by set operations, even the internal representation would have been non-manifold.

- Figure 12 shows the connection of two disconnected toroids. This figure also gives an example of handle improvement by inserting additional edges. As can be seen clearly in Figure 12, when only one edge is inserted it creates a C^1 discontinuous extraordinary vertex. Inserting extra edges removes this C^1 discontinuity.
- Figure 13 shows an example of the creation of a handle for a cup and demonstrates handle improvement by inserting additional edges.
- Figure 14 shows an example of the creation of a hole. As it is clear in this figure, each additional edge improves the quality of the hole in the cube.

The Figures 15, 16, 17 and 18 shows two examples of topological changes. The Figures 15 and 16 shows the creation of various handles for a small stellated dodecahedron [37] (a regular polyhedron discovered by Kepler). The Figures 17 and 18 shows the creation of several holes in a great dodecahedron [37] (A regular polyhedron discovered by DePoinsot, 100 years after Kepler. It is the dual of small stellated dodecahedron).

We offer a few implementation remarks before we close this section.

The topological changes are implemented in our approach by connecting faces in a polygonal mesh. The type of the topological changes is closely related to the visibility of the corresponding

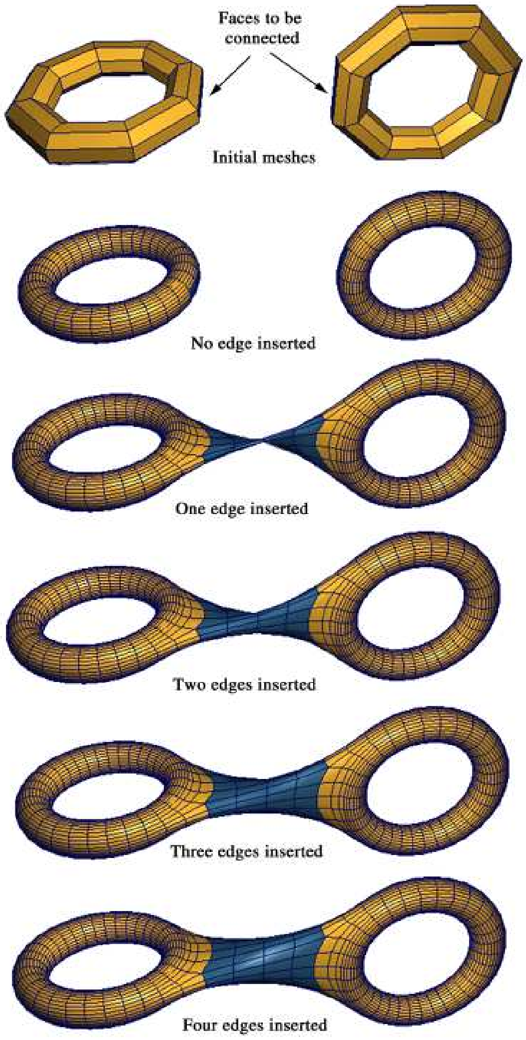

Figure 12: Connecting two toroids: the same colored faces connected with edge insertion.

faces. Generally speaking, if the faces are visible from each other from the "outside" of the mesh, then adding an edge to connect the two faces creates a handle to the mesh. If the faces are visible from each other from "inside" of the mesh, then adding an edge to connect the two faces opens a hole in the mesh. When the two faces are not visible from each other, either from outside or from inside, adding a straight edge to connect the two faces still gives a topologically correct mesh but the mesh will be geometrically self-intersected. Thus, the users can easily avoid such self-intersections based on the face visibility properties.

Using more than one straight edge to connect two faces creates an extraordinary point with very high valance in the face handle since each edge adds 2 to the valance. Such an extraordinary point with a high valance will most likely be C^1 discontinuous. It is easy to avoid extra edges by extruding the faces before inserting an edge as shown in Figure 13 as an example of the creation of a handle.

6 Conclusion and Future Work

In this paper, we presented a new paradigm for changing topology of 2-manifold meshes without using an implicit approach. Our new paradigm guarantees the 2-manifold property for meshes during these topological changes. The new paradigm for changing

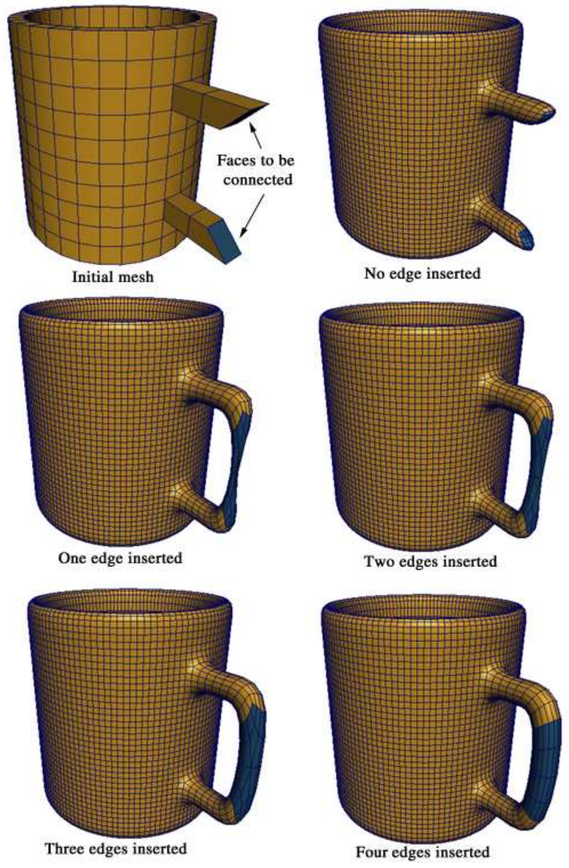

Figure 13: Creation of handle for a cup.

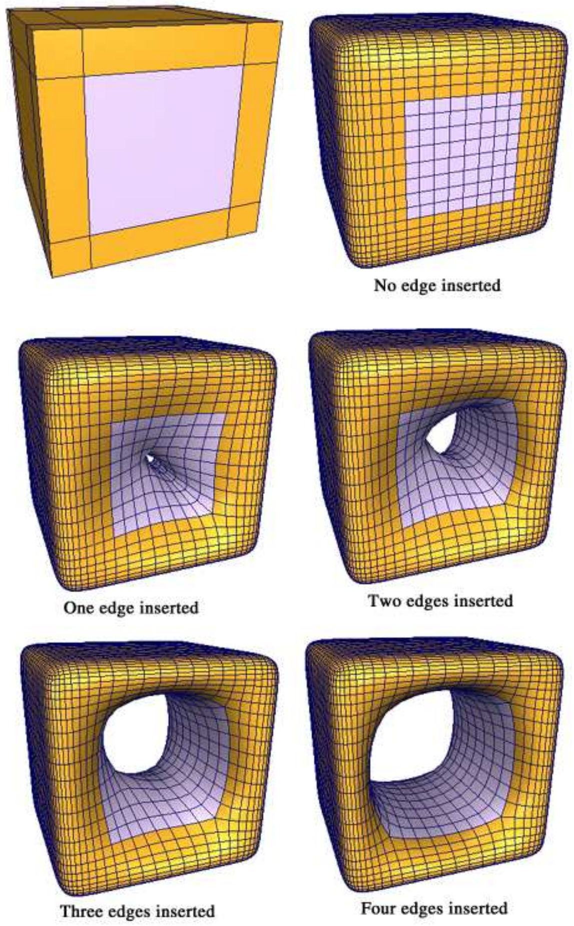

Figure 14: Creation of a hole in a cube.

topology is highly consistent with the subdivision approach in a 3D modeling system. Our approach can be used for modeling initial 2-manifold polygonal meshes, as well as for changing the topology of 2-manifold polygonal meshes that are smoothed by subdivision operations. Note that smoothing operations provided by subdivision approach are essential for improving the quality of 2-manifolds. Thus, the topological change and subdivision operations performed alternatively during model construction provide a powerful shape modeling paradigm that supports a hierarchy of topology changes and quality improvement in different levels of details.

Our approach can also be extended to include topological changes into progressive meshes [24] and multiresolution representations of meshes [13, 9]. Two major operations in progressive meshes technique are vertex-split and edge-collapse. It can be shown that these operations do not change the topology [18]. The same can also be easily shown for multiresolution representations of meshes. In fact, one of the major problem in multiresolution mesh morphing is that source and target must share the same topology [25]. By including the edge insertion and edge deletion operations, it is possible to change the topology for progressive meshes and multiresolution representations. Thus, this approach may potentially add new power to the existing morphing, compression and simplification schemes.

Another direction of extension of the work is to use our approach to model 2-manifolds with boundaries and non-orientable 2-manifolds, such as a projective plane or a Klein bottle [16]. Recent mathematical research [7] has built a solid foundation for using graph rotation systems to represent non-orientable 2-manifolds. A generalized graph rotation system will always guarantee a valid (orientable or non-orientable) 2-manifolds. Our DLFL data structure can also be generalized to include generalized graph rotation systems and, therefore, non-orientable 2-manifolds. The consistency of this approach on non-orientable 2-manifolds and the subdivision algorithms requires further study.

References

[1] By the authors of this paper, "Guaranteeing the 2-Manifold Property for meshes with Doubly Linked Face List", *To appear in a journal*.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, (Addison-Wesley, Reading, MA, 1974).

[3] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, (Morgan Kaufmann Publishers, Los Altos, CA, 1987).

[4] J. Bloomenthal, C. Bajaj, J. Blin, M. P. Cani-gascuel, A. Rockwood, B. Wyvill and G. Wyvill, *Introduction to Implicit Surfaces*, (Morgan Kaufmann Publishers, Los Altos, CA, 1997).

[5] E. Catmull and J. Clark, "Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes", *Computer Aided Design*, **10** (September 1978) 350-355.

[6] J. Chen, D. Archdeacon, and J. Gross, "Maximum Genus and Connectivity", *Discrete Mathematics*, **149** (1996) 19-29.

[7] J. Chen, J. Gross and R. Riper, "Overlap Matrices and Imbedding Distributions", *Discrete Mathematics*, **128** (1994) 73-94.

[8] J. Chen, "Algorithmic Graph Embeddings", *Theoretical Computer Science*, **181** (1997) 247-266.

[9] A. Certain, J. Popovic, T. DeRosa., T. Duchamp, D. Salesin, and W. Stuetzle, "Interactive Multiresolution Surface Viewing", *Computer Graphics*, **30** (August 1996) 91-98.

[10] T. DeRosa. M. Kass and T. Truong, "Subdivision Surfaces in Character Animation", *Computer Graphics*, **32** (August 1998) 85-94.

[11] D. Doo and M. Sabin, "Behavior of Recursive Subdivision Surfaces Near Extraordinary Points", *Computer Aided Design*, **10** (September 1978) 356-360.

[12] J. Edmonds, "A Combinatorial Representation for Polyhedral Surfaces", *Notices American Mathematics Society*, **7** (1960) 646.

[13] M. Eck, T. DeRosa. T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle, "Multiresolution Analysis of Arbitrary Meshes", *Computer Graphics*, **29** (August 1995) 173-182.

[14] E. Ferley, M. P. Cani and J. D. Gascuel, "Practical Volume Sculpting", *Proceedings of Implicit Surfaces'99*, (September 1999) 51-58.

[15] H. Ferguson, A. Rockwood and J. Cox, "Topological Design of Sculptured Surfaces", *Computer Graphics*, **26** (August 1992) 149-156.

[16] P. A. Firby and C. F. Gardiner, *Surface Topology*, (John Wiley & Sons, 1982).

[17] A. T. Fomenko and T. L. Kunii, *Topological Modeling for Visualization*, (Springer-Verlag, New York, 1997).

[18] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, (Wiley Interscience, New York, 1987).

[19] T. A. Galyean and J. F. Hughes, "Sculpting: An Interactive Volumetric Modeling Technique", *Computer Graphics*, **25** (July 1991) 267-274.

[20] M. Halstead, M. Kass, and T. DeRose, "Efficient, Fair Interpolation using Catmull-Clark Surfaces", *Computer Graphics*, **27** (August 1993) 35-44.

[21] L. Heffter, "Uber das Problem der Nachbargebiete", *Math. Annalen*, **38** (1891) 477-508.

[22] C. M. Hoffmann, *Geometric & Solid Modeling, An Introduction*, (Morgan Kaufman Publishers, Inc., San Mateo, Ca., 1989).

[23] C. M. Hoffmann and G. Vanecek, "Fundamental Techniques for Geometric and Solid Modeling", *Manufacturing and Automation Systems: Techniques and Technologies*, **48** (1990) 347-356.

[24] H. Hoppes, "Progressive Meshes", *Computer Graphics*, **30** (August 1996) 99-108.

[25] A. W. F. Lee, D. Dobkin, W. Sweldens and P. Schroder, "Multiresolution Mesh Morphing", *Computer Graphics*, **33** (August 1996) 343-350.

[26] C. Loop, "Smooth Subdivision Surfaces Based on Triangles", Master's Thesis, Department of Mathematics, University of Utah (1987).

[27] C. Loop and T. D. Rosa, "Generalized B-spline Surfaces with Arbitrary Topology", *Computer Graphics*, **24** (August 1991) 101-165.

[28] C. Loop, "Smooth Spline Surfaces over Irregular Meshes", *Computer Graphics*, **28** (August 1994) 303-310.

[29] A. Pasko, V. Adzhiev, A. Sourin and V. Savchenko, "Function Representation in Geometric Modeling: Concepts, Implementations and Applications", *Visual Computer,* no. 11, pp. 429-446, 1995.

[30] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, (Springer-Verlag, 1985).

[31] A. Ricci, "A Constructive Geometry for Computer Graphics", *The Computer Journal*, **16** (May 1973) 157-160.

[32] A. A. G. Requicha and H.B. Voelcker, "Solid modeling: A Historical Summary and Contemporary Assessment", *IEEE Computer Graphics and Applications*, **2** (March 1982) 9-24.

[33] J. Stam, "Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values", *Computer Graphics*, **32** (August 1998) 395-404.

[34] T. W. Sederberg, D. Sewell, and M. Sabin, "Non-Uniform recursive Subdivision Surfaces", *Computer Graphics*, **32** (August 1998) 387-394.

[35] B. T. Stander and J. C. Hart, "Guaranteeing the Topology of an Implicit Surface Polygonization for Interactive Modeling", *Computer Graphics*, **31** (August 1997) 279-286.

[36] S. Takahashi, Y. Shinagawa and T. L. Kunii, "A Feature-Based Approach for Smooth Surfaces", in *Proceedings of Fourth Symposium on Solid Modeling*, (1997) pp. 97-110.

[37] R. Williams, *The Geometrical Foundation of Natural Structures*, (Dover Publications, Inc., 1972).

[38] G. Wyvill, C. McPheeters, and B. Wyvill, "Data Structure for Soft Objects", *The Visual Computer,* vol 2, no. 4, pp. 227-234, 1997.

[39] D. Zorin, editor, *Subdivision for Modeling and Animation,* ACM SIGGRAPH'99 Course Notes no. 37, August, 1999.

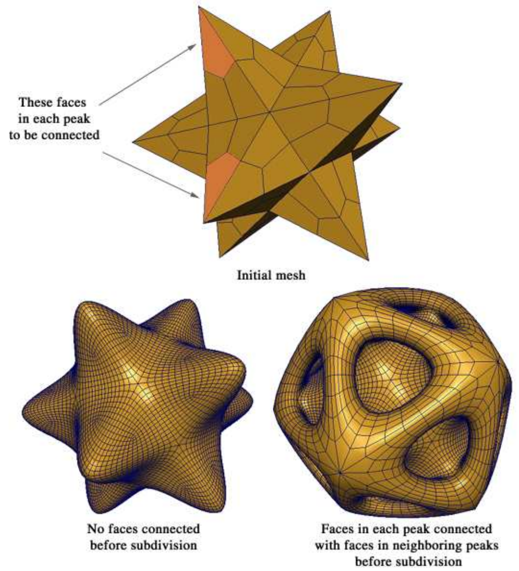

Figure 15: Creation of various handles for small stellated dodecahedron.

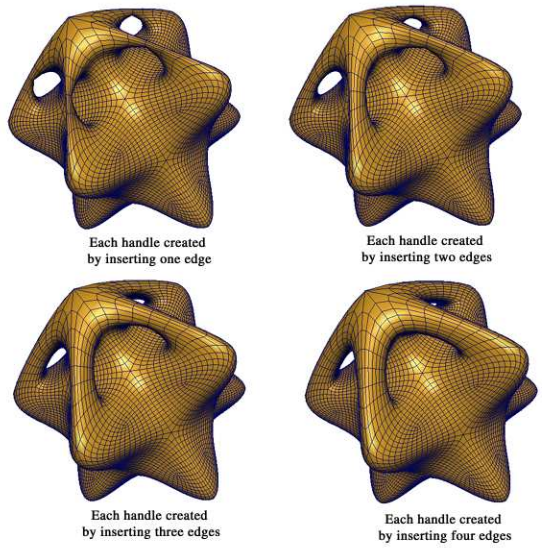

Figure 16: Handles improvement for for small stellated dodecahedron.

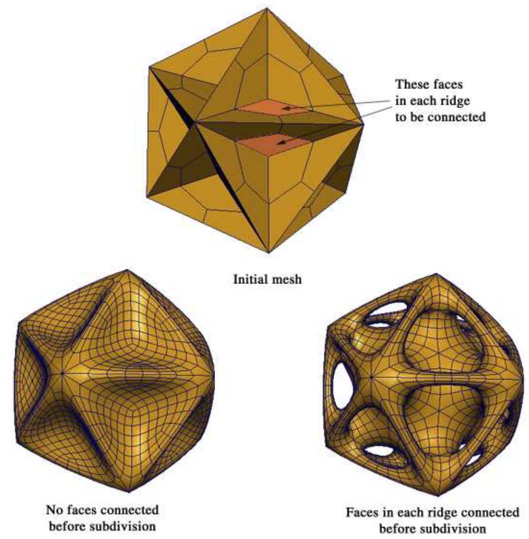

Figure 17: Creation of various holes for great dodecahedron.

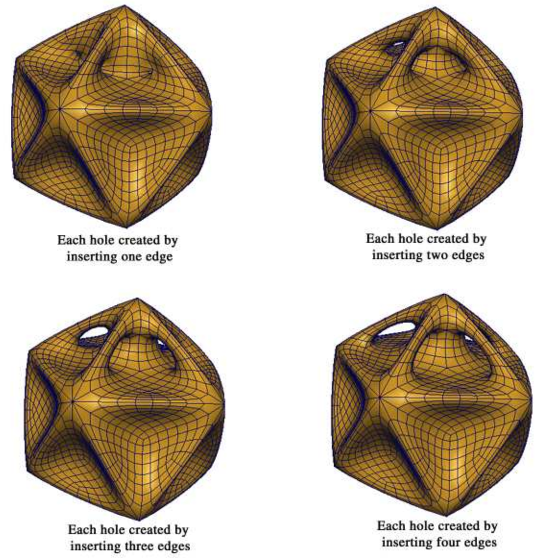

Figure 18: Hole improvement for great dodecahedron.