Topological Morphing Using Reeb Graphs

Pizzanu Kanonchayos* Tomoyuki Nishita* Shinagawa Yoshihisa† Tosiyasu L. Kunii††

**The University of Tokyo* *†University of Illinois at Urbana-Champaign* *††Hosei University*

{pzk, nis}@is.s.u-tokyo.ac.jp *sinagawa@uiuc.edu* *kunii@k.hosei.ac.jp*

Abstract

Metamorphosis between 3D objects is often the transformation between a pair of shapes that have the same topology. This paper presents a new model using Reeb graphs and their contours to create morphing between 3D objects having different topology. The proposed method specifies the correspondence between of the input objects by using the graph isomorphic theory. Then the super Reeb graph, which has the equivalent topological information to the Reeb graphs of the two input objects, is constructed and used to conduct the sequence of the morphing. The evolutions of the topology that occur during the morph can be specified by the transformation of the Reeb graphs and their contours of the input objects. Reeb graph-based modeling allows the users precisely and intuitively control the morph because the topological information of the objects, represented by the structures of the Reeb graphs, is explicit and easy to understand. Moreover, the contours of the Reeb graphs also represent the geometrical information of the objects. The examples of morphing between different topological shapes are demonstrated.

1. Introduction

In recent years, morphing has considerably popularity especially in the entertainment industry. It is also applied not only to the industrial but also to the medical applications such as medical visualization. The success depends on algorithms that allow users to specify correspondences for morphs in an intuitive way. Several algorithms that realized smooth morphing sequences have been proposed.

The task of morphing an object into another object that has different topological information starts from specifying a correspondence between the two objects. It is to specify where features on one object end up on the other object as a result of the transformation. When the topological information changes, the correspondence must also indicate how the change takes place. The morphing engine uses a method of interpolation to effect a transition.

On the 3D mesh morphing, many researches [3, 6] solve the correspondence problem by a sphere or a disk embedded in 3D meshes to find the parameterization correspondence. This method is improved [8] by using a recursive algorithm to find a correspondence. Multiresolution analysis [9] is used to solve the correspondence problem through coarse-to-fine parameterization matching. Combining scaled version of the shapes is used to produce interpolations between shapes by combining scaled version of the shapes [5]. Besides the mesh modeling, morphing on volumetrically sampled implicit surfaces is performed [4]. In this research the smoothness of the transformation is improved by scheduling frequencies using Fourier analysis. However, these methods do not seem to work with the surface meshes that differ topologically.

Some methods of topological evolution of 3D meshes are presented. Intermediated 3D meshes are inserted inbetween the source and the destination meshes to specify the evolution of the topology [1]. All the possible alternatives of topological transitions, however, are not considered and the intermediate shapes are not systematically generated. 4D implicit surface is interpolated between 3D meshes and 3D mesh is morphed by extracting isosurfaces. However, the topological evolution is not explicitly offered. The procedure to explicitly specify the topological evolutions is proposed [14] by using formalism based on the mesh morphing algorithm employing direct interpolation of input 3D meshes by using a 4D tetrahedral mesh. This procedure allows users to control and prevent the ambiguity in the topological evolution. However, the use of the formalism needs skills and knowledge of the topological evolution and may not trivial when the shapes and sizes of the meshes are too complex.

The modeling of Reeb graph-based construction for complex shapes [10, 11, 13, 15] uses Reeb graphs to represent the topology skeletons of 3D objects and cross-sectional contours to represent the surfaces. The advantages for modeling purposes are numerous, from ease of use to compact storage or the ability to be converted from polygonal shapes. Their use in the key-frame animation [7] is also interesting since they can be used to interactively and predictably handle key-frame deformation.

This paper focuses on morphing between topologically different shapes and an automatically specifying the correspondence between the two shapes by using the Reeb graphs and their contours. The corresponding edges and nodes of the Reeb graphs represent the correspondence between two shapes. The transformations of the Reeb graphs specify the topological transition of 3D shapes that must occur during the morph. The Reeb graphs allows the user to explicitly and precisely specify the sequence of topological evolution. The demonstration of the shape morphing is available to show the examples of topological specification.

The remainder of the paper is organized as follow. Section 2 briefly reviews the background on the Reeb graphs. We present our transformations of the Reeb graphs to specify the topological transition during the morph between the objects having different topological information in Section 3. The graph isomorphism algorithm to specify the corresponding nodes and edges of the Reeb graphs is explained in Section 4. Section 5 demonstrates the examples of simple morphing results, and Section 6 concludes this paper and notes the future work.

2. Background on Reeb graphs

In this section, we briefly review the approach for the animation of an object using its Reeb graphs and contours as shown in Figure 1.

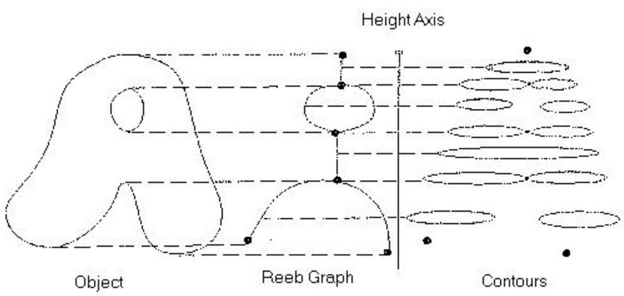

Figure 1. The Reeb graphs and contours of an object.

The Reeb graph represents the topological skeleton of a 3D object and shows between which contours the surface patches should be generated. It is defined by regarding each cross-sectional contour as a point. Adding the contours to represent the geometry of the object enhances the Reeb graphs to be able to represent a 3D object. The object is reconstructed by generating surface patches between the contours. This process uses the homotopy and the continuous toroidal graph [11]. The objects are assumed to satisfy the definition of the Reeb graph based on the height axis. The articulated Reeb graphs are proposed to remedy this problem by allowing each edge of the Reeb graph to hold its local height axis. When the Reeb graph is transformed the edges will be transformed individually. The point-to-point constraint and the re-transformation are used to maintain the connectivity of the Reeb graphs and the proper height axes.

The method used a purely geometric way of transformation without any optimization process to define the skeletons and to generate the surface contours. The deformation of the objects is, hence, predictable because the structures of the Reeb graphs explicitly represent the topological information of the objects. This property, making the method robust to topological changes, seems attractive, especially when used for the generation of metamorphosis between objects that have different topological information. Moreover, this method also provides a solution to the self-intersection problem, which may occur during the transformation.

3. Reeb graph transformation and topology evolution

A smooth transformation between topologically different objects requires both morphing the geometric interpolation and evolution of the topology. As described in the previous section, the Reeb graphs can represent the topological information of the objects while their contours represent the geometrical information. Therefore, the morphing for the geometric interpolation is the morphing between the corresponding contours of the source and the target objects, and the evolution of the topology can be specified by the transformation of the Reeb graphs, which will be detailed later.

To perform the morphing, we give two Reeb graphs G_1 and G_2, which we will refer to as the source and destination Reeb graphs of the input objects that are closed and orientable. In the following sections we may refer only to the transformation or correspondence from G_1 and G_2.

3.1. Topological transformation

The transformation between two topologically different objects involves an evolution of topology. According to the topological transitions [2] arising from the critical points of the surface, which can be invoked by attaching the topological handles to the surface, this allows us to specify eight possible topological transitions [14], which are represented by the following transformations of theReeb Graphs as shown in Figure 2.

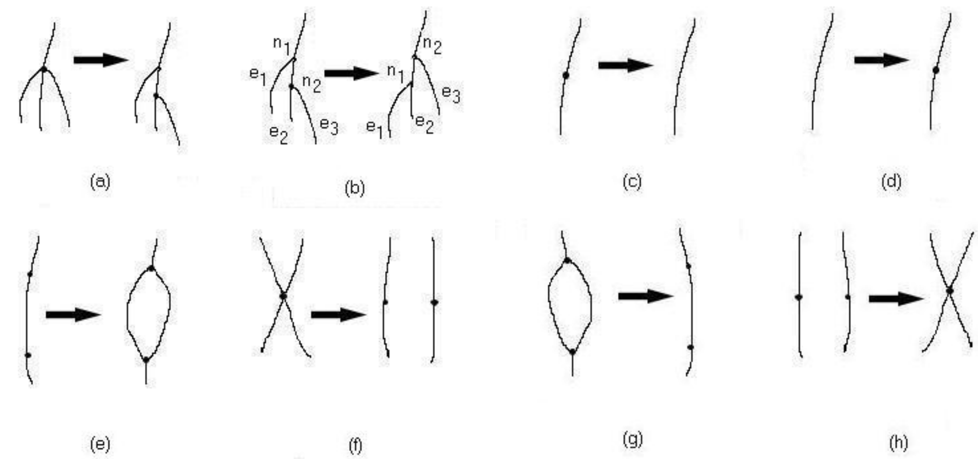

Figure 2. The eight transformations of the Reeb graphs.(a) Transformation 1: Split a one-to-three Reeb node to two one-to-two Reeb nodes. (b) Transformation 2: Shift the right Reeb edge e_3 upward until it is above the left Reeb edge e_1. (c) Transformation 3: Delete a one-to-one Reeb node. (d) Transformation 4: Add a one-to-one Reeb node. (e) Transformation 5: Split a Reeb edge into two. (f) Transformation 6: Split a two-to-two Reeb node into two Reeb edges with one Reeb node. (g) Transformation 7: Join two Reeb edges within a loop into one Reeb edge. (h) Transformation 8: Join two Reeb nodes into one two-to-two Reeb node

.

Transformation 1: Split a one-to-N or N-to-one Reeb node to a collection of articulated one-to-two or two-to-one Reeb nodes as shown in Figure 2(a).

Every one-to-N and N-to-one Reeb node must be split to a collection of articulated one-to-two or two-to-one Reeb nodes and also their contours. This transformation reduces the complexity the structure of the Reeb graph.

Transformation 2: Shift a lower left/right Reeb edge up until it is above the right/left Reeb edge.

A left or right Reeb edge can be shifted up until it is above the Reeb edge in the other side. It means that the collection of articulated one-to-two or two-to-one Reeb nodes may be re-split into another structure. As shown in the Figure 2(b) the Reeb edge e_3 is shifted until it is above the edge e_1 .

Transformation 3: Delete a one-to-one Reeb node and its contour and connect the upper and lower Reeb edges as shown in Figure 2(c).

This transformation does not change the topology of the objects but it is used to ease the structure of the Reeb graph.

Transformation 4: Split a Reeb edges into two Reeb edges and add a Reeb node and its contour between them as shown in Figure 2(d).

This transformation does not change the topology of the objects. The added Reeb node and its contour will be transformed between the corresponding Reeb nodes and the contours of the two Reeb graphs of the input objects.

Transformation 5: Split a Reeb edge between a pair of Reeb nodes into two Reeb edges as shown in Figure 2(e).

This transformation is used to add a hole to the surface.

Transformation 6: Split two-to-two Reeb node and its contour into two one-to-one Reeb nodes and two contours as shown in Figure 2(f).

This transformation is used to split a part of the object into two or merge upper and lower holes into one hole.

Transformation 7: Join two Reeb edges within a loop to one Reeb edge as shown in Figure 2(g).

This transformation is used to merge two parts of the objects into one.

Transformation 8: Join two Reeb nodes and their contours to one Reeb node and one contour as shown in Figure 2(h).

This transformation is used to split a hole into upper and lower holes.

The transformations shown in Figure 2 represent all the possible cases of the topological evolution. The arbitrarily transition is allowed by combining these transformations. Figure 3 shows the transformations of the surfaces of the objects according to the transfomation of the Reeb graphs in Figure 2. However, the objects in this research are fixed in the height axis so it is possible that some topology evolution may not satisfy the definition of the Reeb graphs. The solution is to reconstruct the Reeb graph in a new height axis as described in [7]. In this paper, the user can specify their own evolution of the objects by defining the sequence of the transformations that will be described in Section 4.2.

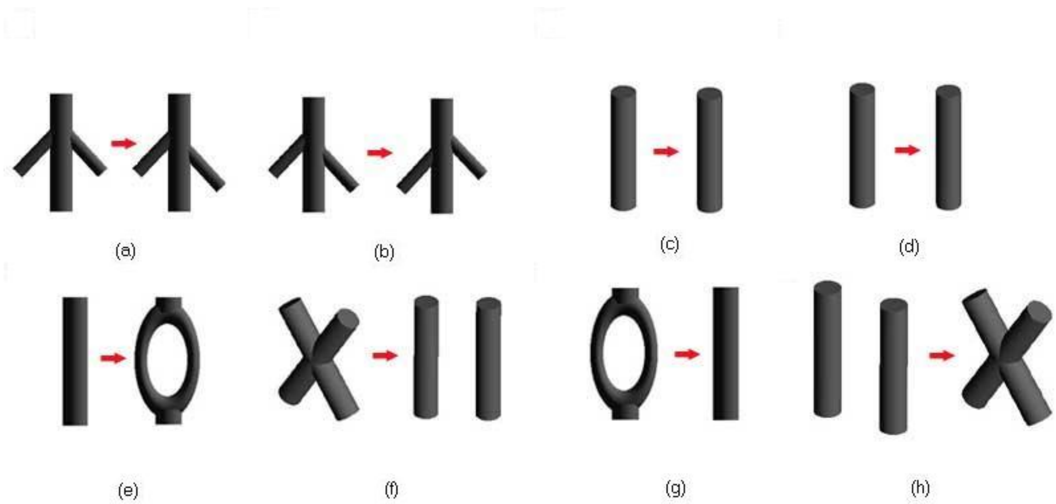

Figure 3. The eight transformations of the surfaces of the objects.(a) Transformation 1: Split a one-to-three Reeb node to two one-to-two Reeb nodes. (b) Transformation 2: Shift the right Reeb edge e_3 upward until it is above the left Reeb edge e_1. (c) Transformation 3: Delete a one-to-one Reeb node. (d) Transformation 4: Add a one-to-one Reeb node. (e) Transformation 5: Split a Reeb edge into two. (f) Transformation 6: Split a two-to-two Reeb node into two Reeb edges with one Reeb node. (g) Transformation 7: Join two Reeb edges within a loop into one Reeb edge. (h) Transformation 8: Join two Reeb nodes into one two-to-two Reeb node.

3.2. Geometry interpolation

In our method the morphing of the geometrical interpolation is the morphing between two 2D contours. A tree structure [11] is employed to efficiently describe the inclusion relations of contours on a 2D plane. The contours of the intermediate Reeb graph are interpolated by homotopy of the input Reeb graphs' contours as shown in Figure 6.

4. Algorithm

Since our method is based on the Reeb graphs considered as directed graphs, we solve the correspondence problem by examining the graph isomorphism problem, which is to find an efficient method for determining if two graphs are isomorphic. In this paper, we use a fast backtracking algorithm [12] that has been effective in testing large classes of directed graphs. This method falls into the class of vertex classification algorithms that are adapted to the class of Reeb node classification algorithm. Then, the digraph isomorphism algorithm [12] provides us with a solution. The answer to the correspondence problem is given when the graphs are isomorphic. Hence, its order is O (n^3) and O (n*n!) in the worst case.

4.1. Super Reeb graphs

Super Reeb graph is an intermediate Reeb graph inbetween the input G_1 and G_2. The Super Reeb graph implements the topological transitions of both input Reeb graphs as in Figure 4. The user can create the source and destination Reeb graphs or input the 3D polygonal objects and let the system convert polyhedra to cross sections and construct the Reeb graphs of the objects. The key of the topological transition in this research is to construct this Super Reeb graph.

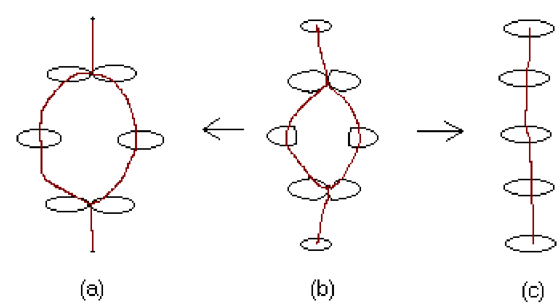

Figure 4. An Example of the Super Reeb graph. (a) Reeb graph G_1 (b) Super Reeb Graph (c) Reeb Graph G_2

The Super Reeb graph must be able to transform to both sources. On the other hand, it must have the topological information equivalent to each object. The Super graph must be a Reeb graph that can be transformed to both G_1 and G_2. Each G_1 and G_2will be transformed to a new Reeb graph that has less complex structure by using the transformation methods described in Section 3.1. The transformation will start with the graph that has more Reeb nodes until the two Reeb graphs have the same node. Then the two transformed Reeb graphs will be checked to see if it is isomorphic or not and the correspondence is specified. If they are equivalent to each other, the process will be terminated and the super Reeb graph is the graph in the final state where the two transformed Reeb graphs are isomorphic. If the two transformed Reeb graphs are not yet isomorphic, the transformations are applied again and again until they are isomorphic and the correspondence is specified. In the worst case, each of the two graphs will be transformed into a one-edge Reeb graph with two contours. That means there is no topological difference between the two transformed Reeb graphs.

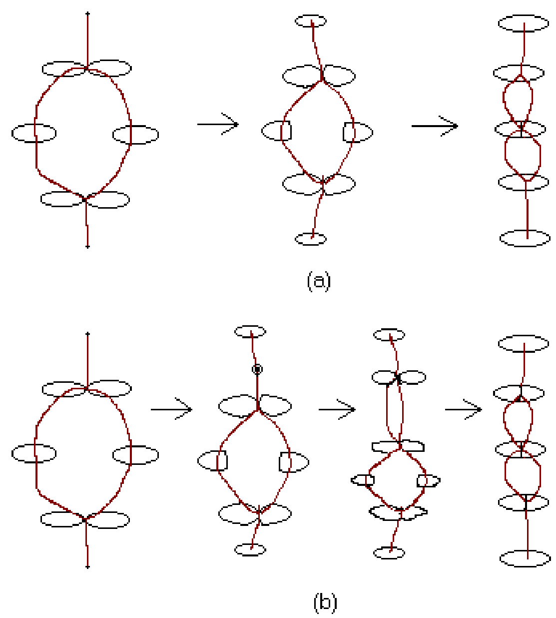

Figure 5. The topological evolution between a torus and an 8-shape Reeb graphs by using different series of the transformations. (a) The morphing by joining two Reeb nodes. (b)The morphing by adding a Reeb node and then splitting an Reeb edge into two.

4.2. Topological transition selection

The user can predefine the sequence of the topological transformation. Then the system will check the correspondence of the input Reeb graphs. If the input Reeb graphs are not topologically equivalent to each other, the input Reeb graphs will be transformed in the order of the sequence that is predefined. Otherwise, the transformations can be selected step by step by the user. The morphing results between Reeb graphs of a torus and a sphere with the different sequence of the transformations are shown in Figure 5. Figure 5(a) is the result of morphing with the sequence of the transformations defined as 1, 2, 3, 4, 5, 6, 7, 8 and Figure 5(b) is the result of the morphing with the sequence of the transformations defined as 1, 2, 3, 4, 8, 7, 6, 5.

5. Experimental Results

This section shows some examples of morphing between 3D objects having different topology. Figure 6 represents the geometrical interpolation between two different objects having the same topology. Figure 7 is a transformation from a torus to a sphere. Figures 8 and 9 are the morphing results between the "0" shape object and an "8" shape object with the different sequences of the Reeb graph transformations as shown in Figures 4 and 5. The transformations proposed in this paper represent the topological transition of the objects. Users can precisely and intuitively select the proper transformation. The ease of the topological representation leads to more predictable and clear morphing result. This method can be also set to the automatic mode to automatically specify the correspondence between the two different topological input objects and give the morphing results according to the predefined transformation sequence, or let the user select the transformations interactively.

6. Conclusions and future works

The main purpose of this paper is to present a method that can produce smooth transformations between the topologically different objects using the Reeb graphs and their contours. Because of the advantage of using the Reeb graph-based model, users can control the transition of the topological information of the objects by selecting the transformations proposed in this paper. The transformation is defined based on the topological transition and can be used to control most possible cases of topological evolution of the 3D objects. The ease and clear structure of the Reeb graph help users who are not familiar with the topology to precisely and intuitively design how the morph should be. The super Reeb graph is introduced to implement the process of morphing. By using the graph isomorphic algorithm, the correspondence between the transformed isomorphic Reeb graphs will be specified. The self-intersection during the transformation is also prevented [7].

As the Reeb graph-based model is fixed to the height axis, the object is considered to be orientable or fixed to the height axis. It means that there may exist some topological transformations that make the Reeb graphs invalid. The height axes may have to be transformed to the proper directions or even the transformed Reeb graphs might need to be reconstructed. Furthermore, the algorithm to specify the correspondence between the input Reeb graphs is still unsatisfactory for complex objects.

We still have to find out the solution for morphing of the open surface objects using the Reeb graphs. Finally, the extension of this method to large, articulated, and more complex models will produce even more interesting results.

References

[1] D. Decarlo, and J. Gallier., "Topological Evolution of Surfaces", *Graphics Interface'96*, 1996, pp. 194-203.

[2] A. T. Fomenko, and T. L. Kunii, *Topological Modeling for Visualization*, chapter 6, Springer-Verlag, 1997, pp. 105-125..

[3] A. Gregory, A. State, M. Lin, D. Monacha, and M. Livingston, "Feature-based Surface Decomposition for Correspondence and Morphing between Polyhedra", *Computer Animation '98*, IEEE Computer Society Press, 1998, pp. 64-71.

[4] J. F. Hughes. Scheduled,"Fourier Volume Morphing, *Proceeding SIGGRAPH'92*, volume 26, July 1992, pp. 43-46.

[5] A. Kaul, and J. Rossignac, "Solid-interpolating Deformations; Construction and Animation of PIPs", *Comput. & Graphics,* 16(1), 1992, pp. 107-116.

[6] T. Kanai, H. Suzuki, and, F. Kimura, "Metamorphosis of Arbitrary Triangular Mesh", *IEEE Computer Graphics and Applications*, 20(2), 2000, pp. 62-75.

[7] P. Kanongchaiyos, and Y. Shinagawa, "Articulated Reeb Graphs for Interactive Skeleton Animation",*Proceeding Modeling Multimedia Information and System*", 2000, pp. 451-467.

[8] R. E. Parent, "Shape Transformation by Boundary Representation Interpolation: A Recursive Approach to Establishing Face Correspondences", *Journal of Visualization and computer Animation*, volume 3, 1992, pp. 219-239.

[9] A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schroder, "Multiresolution Mesh Morphing", *Computer Graphics (Proceedings Siggraph '99)*, 1999, pp. 343-350.

[10] Y. Shinagawa, T. L. Kunii, Y. Nomura, T. Okuno, and Y. H. Young, "Automating View Function Generation for Walkthrough Animation Using A Reeb Graph", *Proceeding of Computer Animation' 90*, 1990, pp. 227-237.

[11] Y. Shinagawa, and T. L. Kunii"The Homotopy Model: A Generalized Model for Smooth Surface Generation from Cross Sectional Data", *Visual Computer,* 7, 1991, pp. 72-86.

[12] D. C. Schmidt, and L. E. Druffel, "A Fast Backtracking Algorithm to Test Directed Graphs for Isomorphism Using Distance Matrices", *J. A. C. M.*, 23, 1976, pp. 433-445.

[13] C-L. Tai, Y. Shinagawa, and T. L. Kunii, "A Reeb Graph-based Representation or Non-sequential Construction of Topologically Complex Shapes", *Computer & Graphics*, 22, 1998, pp. 255-268.

[14] S. Takahashi, Y. Kokojima, and R.Ohbuchi,"Explicit control of Topological Transitions in Morphing Shapes of 3D meshes", *Proceedings of the ninth Pacific Conference on Computer Graphics and Applications (PACIFIC GRAPHICS-01)*, IEEE Computer Society, October 16-18, 2001, pp. 70-81.

[15] S. Takahashi, and T. L. Kunii, "Manifold-based Multiple Viewpoint CAD – A Case Study of Mountain Guide Map Generation", *Computer Aided Design*, 1994, pp. 622-631.

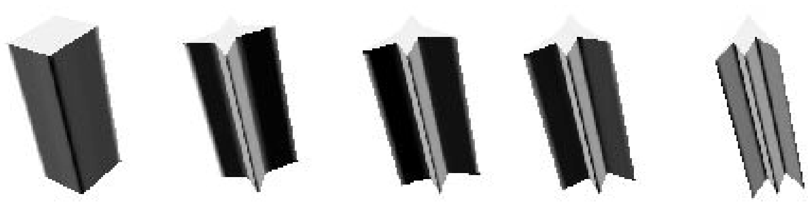

Figure 6. The geometrical evolution between two objects having the same topology.

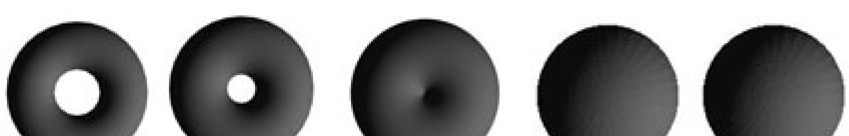

Figure 7. The topological evolutions between a torus and a sphere.

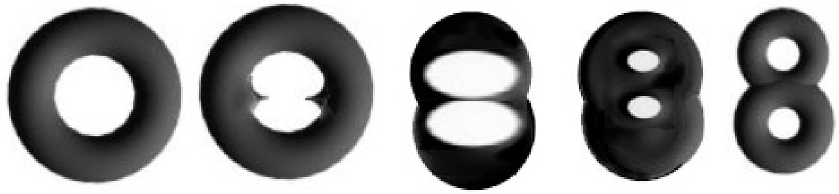

Figure 8. The topological evolution between a 0-shape object and an 8-shape object by using the transformation sequence: 1, 2, 3, 4, 5, 6, 7, 8.

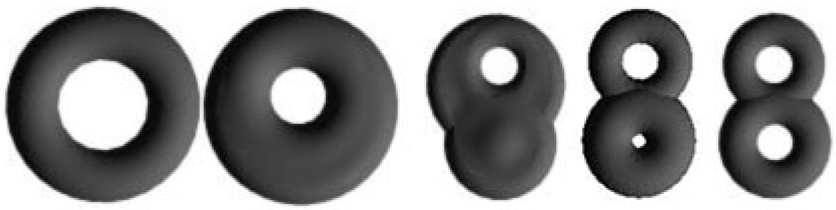

Figure 9. The topological evolution between a 0-shape object and an 8-shape object by using the transformation sequence : 1, 2, 3, 4, 8, 7, 6, 5.