Explicit Control of Topological Transitions in Morphing Shapes of 3D Meshes

Shigeo Takahashi
Department of Graphics and Computer Science
Graduate School of Arts and Sciences
University of Tokyo
takahashis@acm.org

Yoshiyuki Kokojima
Corporate Research and Development Center
Toshiba Corporation
yoshiyuki.kokojima@toshiba.co.jp

Ryutarou Ohbuchi
Computer Science Department
Faculty of Engineering
Yamanashi University
ohbuchi@acm.org

Abstract

Existing methods of morphing 3D meshes are often limited to cases in which 3D input meshes to be morphed are topologically equivalent. This paper presents a new method for morphing 3D meshes having different surface topological types. The most significant feature of the method is that it allows explicit control of topological transitions that occur during the morph. Transitions of topological types are specified by means of a compact formalism that resulted from a rigorous examination of singularities of 4D hypersurfaces and embeddings of meshes in 3D space. Using the formalism, every plausible path of topological transitions can be classified into a small set of cases. In order to guide a topological transition during the morph, our method employs a key-frame that binds two distinct surface topological types. The key-frame consists of a pair of "faces", each of which is homeomorphic to one of the source (input) 3D meshes. Interpolating the source meshes and the key-frame by using a tetrahedral 4D mesh and then intersecting the interpolating mesh with another 4D hypersurface creates a morphed 3D mesh. We demonstrate the power of our methodology by using several examples of topology transcending morphing.

Keywords: *3D mesh morphing, topological evolutions, 4D hypersurfaces, tetrahedral meshes, critical points, topological handles, embeddings, key-frames*

1. Introduction

Recent movies, TV advertisements, and computer games have succeeded in generating astonishing visual effects by using a shape blending technique called *morphing*. The morphing technique is applied to entertainment, industrial, and medical applications, for example 3D contents design and medical visualization. Shape morphing has been studied since around 1980s, and many algorithms have been devised especially for morphing 2D images. Recently, 3D shape morphing has gained special attention due to its ability to produce stunning visual effects. Several algorithms realized smooth morphing sequences of 3D meshes.

One of the key problems in 3D mesh morphing is a correspondence problem, which is defined to be a problem of how to make a parametrization correspondence between a source and a destination 3D meshes. Gregory et al. [3] and Kanai et al. [4] successfully solved this problem by embedding input 3D meshes into a sphere or a disk and then finding the parametrization correspondence there. Furthermore, Lee et al [5] used the multiresolution analysis in order to solve the correspondence problem via coarse-to-fine parametrization matching. However, these methods are limited to cases in which 3D input meshes to be morphed are topologically equivalent (i.e., homeomorphic).

To our knowledge, only a few methods have been proposed that try to morph between shapes having different 3D surface topology.

DeCarlo and Gallier [1] presented a method of specifying the topological evolution of 3D meshes, by inserting intermediate 3D meshes in between the input (i.e., source and destination) meshes. However, they did not consider all the possible alternatives of topological transitions, and

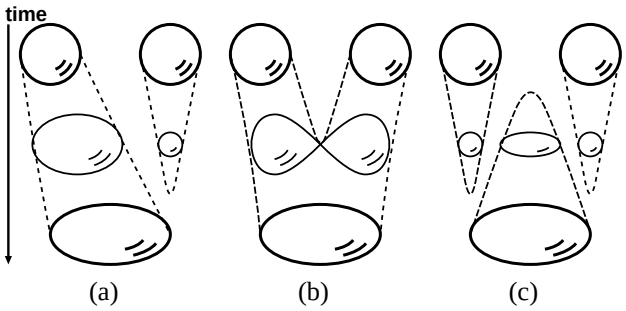


Figure 1. Ambiguity in topological evolution: Morphing from two spheres to one sphere.

their intermediate shapes are generated in an ad hoc manner without a systematic model. Turk and O'Brien [13] considered a 4D implicit surface that directly interpolated between input 3D meshes and then generated a 3D mesh morphing sequence by extracting isosurfaces with respect to the time. While their method can automatically produce topological evolution between any pair of 3D meshes, it does not offer any way to explicitly specify the topological evolution out of possible candidates through user interaction.

Figure 1 shows several alternative ways to morph between two spheres into one sphere; (a) the right sphere disappears while the left one survives, (b) two spheres are joined to become a single sphere, and (c) both of the two spheres disappear and a new sphere appears. As evidenced in the examples of this figure, it is not possible to uniquely determine the topological evolution only from the end (i.e., the source and the destination) 3D meshes. Thus, a mechanism to explicitly control such topological evolutions is called for. Furthermore, the mechanism should be founded on a rigorous mathematical model of surface topology so that the model rules out such anomalies as self-intersection in evolving mesh surfaces.

This paper presents a new method for morphing 3D meshes having different surface topological types. The foremost contribution of this paper is a compact formalism to explicitly and precisely specify the type of topological transition of 3D shapes that must occur during the morph. The paper also presents an implementation of the formalism, which is based on our previous mesh morphing algorithm [8] that employed direct interpolation of input 3D meshes by using a 4D tetrahedral mesh. We demonstrate the beauty and power of the formalism and its realization by examples of topology transcending shape morphing.

In this paper, we assume that input 3D meshes are *closed* and *orientable*, and hence can be embedded in 3D space without any self-intersections. This assumption excludes unusual surfaces such as Möbius strips, projective planes, and Klein bottles from the subject of the present paper, and allows us to enumerate all the acceptable topological transitions of 3D meshes.

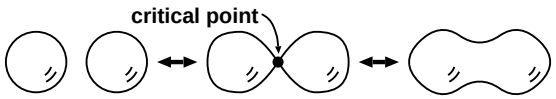


Figure 2. A critical point of a 4D hypersurface shown as a set of its 3D projections.

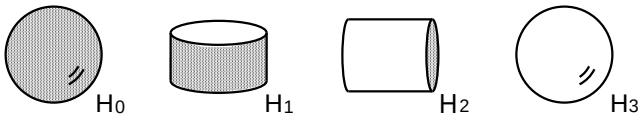


Figure 3. Four topological handles.

This paper is organized as follows. Section 2 presents the complete list of all the possible topological transitions during the morph between 3D meshes having different topological types. This list comes from a rigorous investigation of singularity theory for 4D hypersurfaces. Our new morphing algorithm based on this investigation is described in Section 3 where we detail how to explicitly control the type of topological transition of 3D meshes. Section 4 demonstrates the power of the present methodology by using several simple morphing results, and Section 5 concludes this paper and refers to future work.

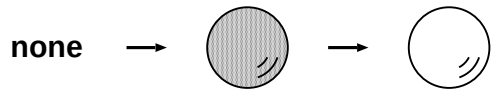
2. Classification of Topological Transitions

As described earlier, our morphing algorithm directly interpolates given 3D meshes by using a 4D tetrahedral mesh (i.e., discretized version of 4D smooth hypersurface), which is embedded in 4D space spanned by x, y, z, and t (or time)-axes [8]. Here, a *critical point* of a 4D hypersurface is defined as a point where isosurfaces with respect to time (t) split or join as shown in Figure 2. If a topological evolution occurs during the morph, it occurs at a critical point of the interpolating 4D hypersurface. Rigorous examination of the behavior of the 4D hypersurface around a critical point resulted in a simple classification of all the possible topological transitions.

According to the Morse lemma, the shape of an infinitesimal neighborhood of the 4D hypersurface around a critical point is categorized into one of the four quadratic forms of f as its local coordinates if the critical point is non-degenerate:

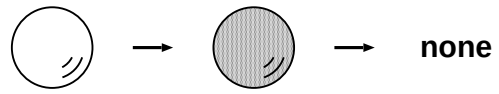
$$f = \begin{cases} -x^2 - y^2 - z^2 & C_3 \ \text{(index 3)} \\ -x^2 - y^2 + z^2 & C_2 \ \text{(index 2)} \\ -x^2 + y^2 + z^2 & C_1 \ \text{(index 1)} \\ x^2 + y^2 + z^2 & C_0 \ \text{(index 0).} \end{cases} \quad (1)$$

Here, the index represents the number of negative eigenvalues of the Hessian matrix at the corresponding critical point. This allows us to classify the critical points of a 4D hyper-



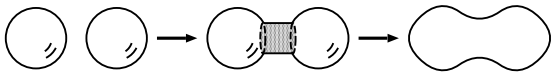
appearance

Figure 4. Topological transition invoked by a topological handle H_3.

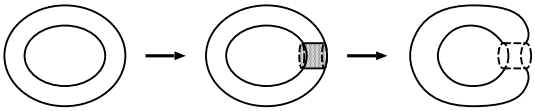


disappearance

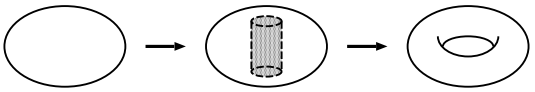
Figure 5. Topological transition invoked by a topological handle H_0.



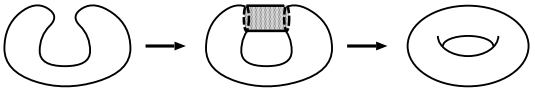
(a) Merges two isosurfaces into one.



(b) Opens a hole to an internal void.

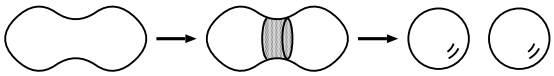


(c) Opens a through-hole.

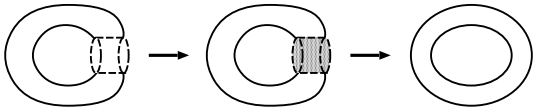


(d) Makes a ring.

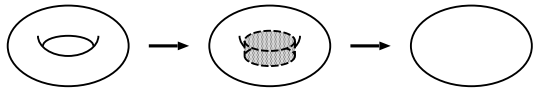
Figure 6. Topological transitions invoked a the topological handle H_2.



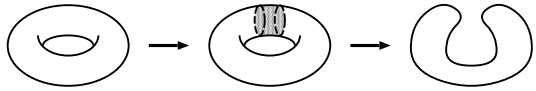
(a) Splits an isosurface into two.



(b) Closes a hole to create an internal void.



(c) Closes a through-hole.



(d) Cuts a ring.

Figure 7. Topological transitions invoked by a topological handle H_1.

surface into four types according to their indices: C_3 (index 3), C_2 (index 2), C_1 (index 1), and C_0 (index 0).

Fomenko and Kunii described in their book [2] that topological transition arising from these critical points can be invoked by attaching one of four *topological handles* to existing isosurfaces. Figure 3 illustrates the four types of the topological handles H_3, H_2, H_1, and H_0, each of which corresponds to C_3, C_2, C_1, and C_0, respectively. For example, the topological handle H_3 invokes an appearance of a new isosurface as shown in Figure 4. Conversely, the topological handle H_0 makes an existing isosurface disappear as shown in Figure 5.

The topological handles H_2 and H_1 generate more complex behavior when the embeddings of time-varying isosurface in 3D space are taken into account. In the case of 2D isocontours, Shinagawa et al. [10] employed a tree structure to effectively describes the inclusion relations of contours on 2D plane in their 3D surface coding system. Our algorithm extends their framework for describing the embeddings to the case having one higher dimension, that is, the inclusion relation of time-varying isosurfaces in 3D space.

We begin by examining the behavior of the topological handle H_2. The topological handle H_2 always merges two isosurface regions into one as shown in Figure 6. Figures 6(a) and (b) show the cases where the two regions are disconnected while Figures 6(c) and (d) show the cases where the two regions lie on the same connected component. These two groups can be classified further depending on the embeddings of isosurfaces in 3D space. The former two cases are classified by considering tree representations of the isosurfaces' inclusion relations. Expressed in inclusion relation trees, the two isosurfaces are sibling in Figure 6(a) while a parent and a child in Figure 6(b). (Note in Figure 6(b) that the two concentric circles in the figure denote two concentric spheres in 3D.) The latter two cases are classified depending on whether inside or outside of the isosurface the topological handle H_2 is attached. In Figure 6(c), the topological handle H_2 melds together the two inner faces of an isosurface, and in Figure 6(d) it melds together two outer faces. The topological handle H_2 yields

the total of four topological transitions shown in Figure 6.

Conversely, the topological handle H_1 reverts the topological transitions caused by the handle H_2, i.e., it splits one isosurface region into two. As a result, H_1 induces the four topological transitions as shown in Figure 7.

The classification illustrated in Figures 4–7 enumerates all the possible cases of topological evolution. An arbitrarily complex topological transition can be created by combining these ten simple cases of topological transitions invoked by the four topological handles. Recall that we assumed any time-varying isosurface to be orientable and hence be embeddable in 3D space.

Topological transitions may include degenerate critical points, for example, the case in which more than two spheres simultaneously adjoin at a discrete point, and the case in which a pair of spheres adjoins at a line segment or a at a finite area. Our present classification of topological transitions fundamentally assumes a 4D interpolator hypersurface having only non-degenerate critical points. This assumption is not a critical limitation, however. Any degeneracy can be handled by our formalism by first decomposing the degeneracy into a sequence of non-degenerate topological transitions that occur at non-zero time intervals and then reducing the time intervals to zero. (Details of this process can be found in [9] for the case of 3D surfaces.) Thus, in the following, we will found our formalism on the assumption of non-degenerate critical points.

3. Algorithm

Based on the formalism described in the previous section, we extend the shape morphing algorithm by Ohbuchi et al. [8] so that transitions of surface topology can be controlled explicitly while morphing 3D meshes that have different topological types. Procedurally, a new stage to generate what we call *key-frame* is added to their shape morphing algorithm, as shown in Figure 8. The key-frame, which will be detailed later, is the tool we devised to realize the effect caused by the four topological handles. In this section, we present the overview of our new shape morphing algorithm. We briefly review Ohbuchi et al's shape morphing algorithm described in [8] as well as the new step added for the explicit control of topology during a topology transcending morphing.

Our updated shape morphing algorithm consists of the following six steps (See Figure 8.):

1. *Multiresolution analysis*
2. *Key-frame generation*
3. *4D base tetrahedral mesh generation*
4. *4D tetrahedral mesh refinement*
5. *4D tetrahedral mesh smoothing*
6. *3D time-varying isosurface extraction*

The second step, key-frame generation, is the step added to explicitly control topological transitions. Each step will be explained in more detail in the subsequent subsections. In the rest of this paper, we denote two input 3D meshes by M_s (the source mesh) and M_d (the destination mesh), and the fourth dimension, time, by t.

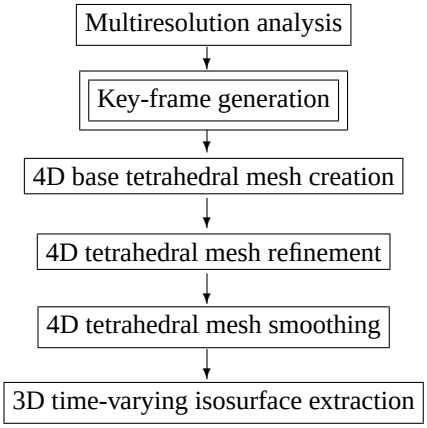


Figure 8. Flow chart of our algorithm; The key-generation step is newly introduced to the algorithm.

3.1. Multiresolution Analysis

The input 3D meshes are wavelet-analyzed to construct a pyramidal hierarchy called a *multiresolution representation* using the framework of Lounsbery et al. [7]. The analysis first simplifies each 3D mesh to find a coarse mesh that approximates the original shape of the input mesh, and then reparametrizes the simplified mesh to construct a coarse-to-fine mesh hierarchy. Since the resultant hierarchy requires 1-to-4 subdivision connectivity, we employed the MAPS algorithm proposed by Lee et al. [6] for the task.

Figure 9 shows examples of multiresolution representations of a torus (M_s) and a sphere (M_d), in which M_s^i (M_d^i) represents a reparametrized mesh of M_s (M_d) at resolution level i. Here, meshes at level 0 (i.e., M_s^0 and M_t^0) serve as a basis for mesh refinement at higher resolution levels 1, 2, and 3. The refined mesh at the highest resolution level is the remeshed version of the input 3D mesh.

As we will discuss later in Sections 3.3–3.5, our algorithm first constructs a 4D interpolating tetrahedral mesh between the coarsest 3D meshes, that are M_s^0 and M_d^0, and then refines and smoothes the tetrahedral mesh to create a smooth 3D mesh morphing sequence. This approach, which applies wavelet-based multiresolution analysis both to the interpolating tetrahedral mesh and to the input (i.e., source and destination) meshes, has two advantages. First, we only need to construct the key-frame at the coarsest resolution

level to bridge the input meshes at that resolution level, that are, M_s^0 and M_d^0. The meshes M_s^0 and M_d^0 retain surface topology of the original input meshes M_s and M_d despite their simplified vertex connectivity and geometry. The key-frame designed at the coarsest resolution level performs the task of uniquely bridging disparate topology of the input meshes. Details of the key-frame generation will be discussed in Section 3.2. Second, the multiresolution hierarchy also allows us to modify the shapes of the interpolating 4D tetrahedral mesh by using different geometric constraints attached simultaneously at multiple resolution levels of the interpolator tetrahedral mesh. The constraint, solved during the mesh smoothing step (Step 5) [11], allows us to introduce various shape morphing effects, such as exaggerated or spatially non-uniform shape transitions.

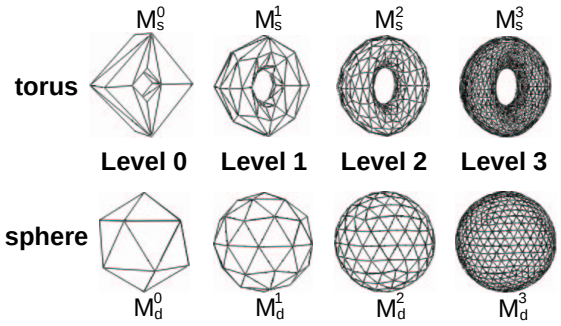


Figure 9. Multiresolution representations of a torus and a sphere.

3.2. Key-frame Generation

The goal of this research is to explicitly and precisely control a topological transition (or transitions) that should occur during a morph. To achieve this goal, our algorithm inserts an intermediate shape called a *key-frame* in between the input 3D meshes at the point, both in time and space, where the topological transition should take place. The key-frame plays an essential role in implementing the topological transitions enumerated in Figures 4–7 caused by the four topological handles H_0–H_3.

The key-frame has two "faces", each of which has the topological type equivalent to one of the two input 3D meshes to be morphed. Here, the two "faces" are implemented as 3D meshes called *key-meshes*, denoted by K_s and K_d, which are homeomorphic, respectively, to the source and destination meshes M_s^0 and M_d^0. Note here that the key-meshes K_s and K_d have the same vertex coordinates and hence the same geometric shape, despite their discrepancy in surface topology (i.e., vertex connectivity). Examples of the key-meshes will be presented later in Figure 11. This dual-faced key-frame bridges the topological difference of the two input meshes M_s^0 and M_d^0. A key-frame is an embodiment of a topological handle pasting operation. The two faces (i.e., key-meshes) of the key-frame represent two states, the states before and after the topological transition caused by a topological handle.

Let us consider the example of morphing a torus (a source mesh M_s^0) into a sphere (a destination mesh M_d^0). The key-frame is generated by the steps below:

1. Topological transition selection:

   Desired topological transition is selected from the possible candidates of Figures 4–7. The morphing between a torus and a sphere involves the topological transition shown in Figure 7(c).

2. Topological handle design:

   A region to which the corresponding topological handle is pasted is selected. This is done by picking vertices bounding the closed surface area that adjoins the topological handle. Figure 10 shows, in gray, the region and its corresponding topological handle H_1 to be pasted together. The geometry of the topological handle is generated so that the handle fills the hole of the torus.

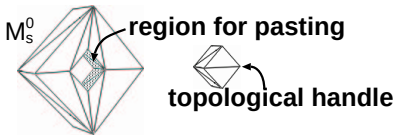


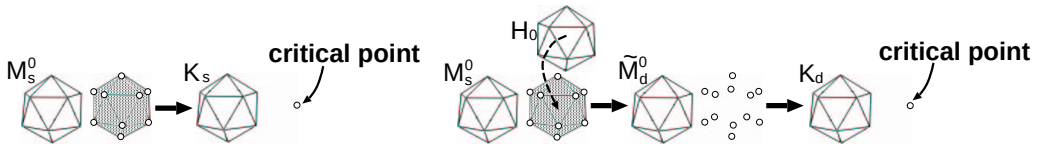
Figure 10. A topological handle and a region where it is pasted.

3. Critical point positioning:

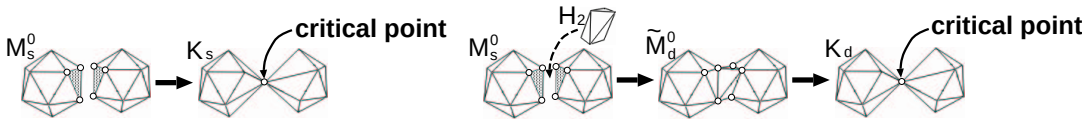
   This step takes user-specified 4D coordinates of the critical point, the spatial point at which the topological transition occurs. We can specify its 3D (x, y, z)-coordinates as the barycenter of the vertices contained in the region for pasting the topological handle. The t-coordinate is set, by default, at the midpoint (in the time axis) between the source and destination 3D meshes. That is, our algorithm defines the time interval for morphing as $[0.0, 1.0]$ and sets the t-coordinate to 0.5. Note that the t-coordinate of the critical point is free to move during the smoothing step later (cf. Section 3.5) while its spatial (3D) coordinates are fixed.

4. Source key-mesh (K_s) generation:

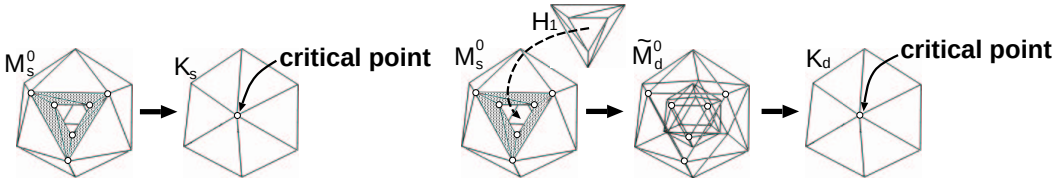
   The coordinates of the vertices in the pasting region are set to those of the critical point specified above. The left-hand side of Figure 11(d) shows an example of this operation. The resultant mesh serves as one "face" of the key-frame, i.e, the source key-mesh K_s. Since the source key-mesh K_s and a torus (M_s^0) are topologically equivalent, we can easily establish the tetrahedral mesh for interpolation between them.



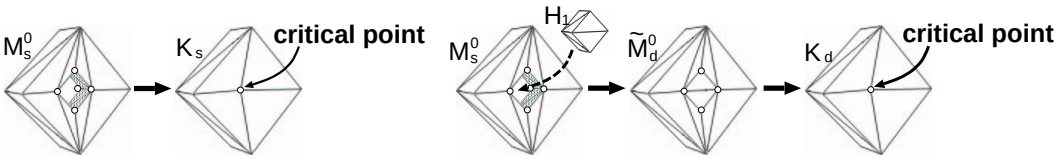
(a) A key-frame to make one of the two isosurfaces disappears (cf. Figure 5).



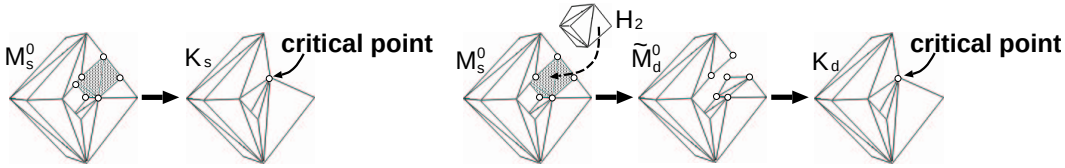
(b) A key-frame to join two disconnected isosurfaces (cf. Figure 6(a)).



(c) A key-frame to close an opening into a void within a solid (cf. Figure 7(b)).



(d) A key-frame to plug a hole of a torus (cf. Figure 7(c)).



(e) A key-frame to cut the ring of a torus (cf. Figure 7(d)).

Figure 11. Examples of key-frames, each of which consists of a pair of key-meshes K_s and K_d.

5. Destination key-mesh (K_d) generation:

   Predefined topological handle is attached to the source mesh M_s^0, and the area to which the handle is pasted is eliminated. This produces a new mesh $\tilde{M}_d^0$ that is topologically equivalent to the destination mesh M_d^0. Again, the coordinates of the vertices in the pasting region are set to those of the critical point. In the case of the example of morphing a torus into a sphere, the right-hand side of Figure 11(d) depicts this operation.

Note that, although the two faces of the key-meshes K_s and K_d have the same geometrical coordinates, they have different surface topology. In this example, while the source key-mesh K_s and the destination key-mesh K_d look the same, they are topologically different. The K_s has a size zero hole of a torus while the K_d has no hole for the hole was plugged with the topological handle H_1. The key-frame thus uniquely bridges the difference in surface topological types between the torus and the sphere, so that topological transition of Figure 7(c) is uniquely and unambiguously selected out of the four alternatives that could be caused by the handle H_1 (cf. Figure 7). Without the precise specification, any combinations of the ten topological transitions in Figures 4–7 might happen.

A key-frame generated through the similar step could unambiguously cause any one of the topological transitions listed in Figures 4–7. For example, the topological transitions of Figures 5, 6(a), 7(b), and 7(d) are realized by generating the topological handles shown in Figures 11(a), (b), (c), and (e), respectively.

It should be emphasized here that the geometric coordinates of the key-frame have little influence on the resultant morphing sequences because its vertex coordinates are variationally optimized by using smooth filtering as described in Section 3.5. It is thus sufficient to design the geometric shapes of the key-frames only approximately.

3.3. 4D Base Tetrahedral Mesh Generation

Our morphing algorithm directly interpolates input 3D meshes by using a 4D tetrahedral mesh in 4D space of x, y, z, and t (time). To accomplish this, the base level (that is, coarsest resolution) input 3D meshes M_s^0 and M_d^0 are placed so that they sandwich the key-frame in the time axis. As discussed in the previous section, the key-frame is a single object with two "faces" K_s and K_d, each of which matches the surface topology of each of the two input meshes. 4D tetrahedra of the interpolator hyper-surface are stuffed in between the meshes M_s^0 and K_s and the meshes M_d^0 and K_d.

In the example of morphing a torus M_s^0 into a sphere M_d^0, Figure 12 illustrates how the source, key-frame, and the destination meshes are lined up in time. Figure 13 shows the initial interpolator tetrahedral mesh created by stuffing tetrahedra in between the source, key-frame, and the destination meshes. A "face" K_s of the key-frame is topologically equivalent to the base-level source mesh M_s^0, and another "face" K_d of the key-frame is topologically equivalent to M_d^0. Thus, tetrahedra can be stuffed between the meshes M_s^0 and K_s as well as the meshes K_d and M_d^0 to create the tetrahedral interpolator surface as shown in Figure 13.

This "dual-face" key-frame approach enabled our new algorithm to morph meshes having different topological types within the framework of our previous shape morphing algorithm [8] that could only morph between 3D meshes having the same topological type.

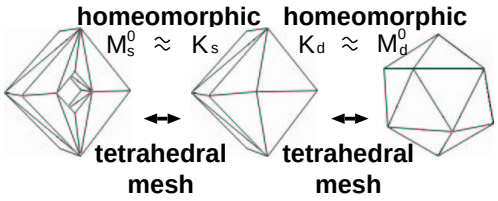


Figure 12. Stuffing tetrahedra in between 3D meshes and a key-frame.

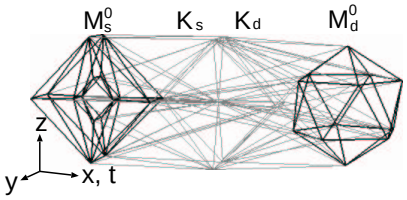


Figure 13. Initial interpolation between base 3D meshes and a key-frame by using a 4D tetrahedral mesh.

3.4. 4D Tetrahedral Mesh Refinement

The base tetrahedral mesh, which has only a small number of vertices, is limited to representing a very coarse morphed 3D shapes. In order to smoothly approximate complex morphing sequences of 3D meshes, our algorithm increases the degrees-of-freedom of the tetrahedral mesh by applying 1-to-8 subdivision rule as shown in Figure 14 to every tetrahedron involved.

Figure 15 shows a tetrahedral mesh obtained by refining the base tetrahedral mesh of Figure 13 three times. Note that, with each subdivision of the tetrahedral interpolator mesh, input meshes M_s and M_d are also refined so that their vertex coordinates become identical to those obtained by the multiresolution analysis step described in Section 3.1. For example, the tetrahedral mesh of Figure 15 contains a reparametrized torus M_s^3 and sphere M_d^3 at its ends. Af-

ter the smoothing step described in the next section, the increased degrees-of-freedom of the tetrahedral mesh allows us to smoothly interpolate the original 3D meshes and the key-frame.

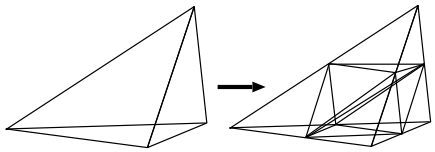


Figure 14. The 1-to-8 symmetrical subdivision rule for tetrahedra.

3.5. 4D Tetrahedral Mesh Smoothing

The refined and subdivided tetrahedral mesh is variationally optimized so that it can produce a smooth morphing sequence of 3D meshes. The variational optimization is implemented as an iterative method using a 4D version of Gaussian-based smoothing filter [12]. The variational optimization treats the vertex coordinates of the critical point and the input 3D meshes M_s^i and M_t^i (at resolution level i) as geometric constraints. The user may impose additional 4D geometric constraints, such as points and curves embedded in 4D space, on the tetrahedral mesh. Thanks to the multiresolution representation framework, these additional constraints may be added at multiple resolution levels [11] to manipulate the shapes of the interpolator surface. These user-specified constraints are used to impose feature correspondence and to create such morphing effects as spatially non-uniform shape transitions [8].

3.6. 3D Time-varying Isosurface Extraction

In this final step, the resultant interpolator tetrahedral mesh is intersected with another 4D hypersurface to extract an interpolated 3D mesh. This is accomplished by finding the intersections between the 4D surface and each tetrahedron included in the tetrahedral mesh. The intersection becomes a set of triangles which in fact is the interpolated (morphed) 3D mesh.

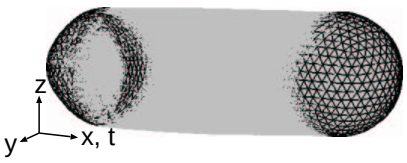


Figure 15. 4D tetrahedral mesh at level 3.

4. Experimental Results

This section presents several examples of morphing between 3D meshes having different topological types. The topological transitions are explicitly controlled by using the method presented in this paper. Our system is implemented on a PC platform (Intel Pentium III CPU 700 MHz with 512 MB memory), and the computation time for each of the following examples took about 10 minutes. This computation includes all the shape morphing steps, i.e., reparametrization, initial meshing, refinements, smoothing, and shape extraction (cf. Sections 3.1, 3.3–3.6), except for the key-frame generation that requires user interaction (cf. Section 3.2).

Figures 16(a), (b), (c), (e), and (f) present the morphing between 3D meshes having different topological types, which involve the topological transitions of Figures 5, 6(b), 7(b), 7(c), and 7(d), respectively.

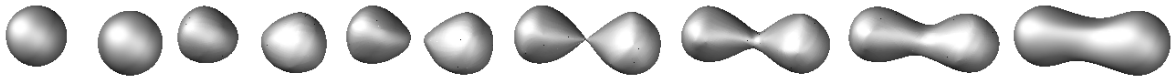
Note that the topological transitions of Figures 16(a) and (b) realize the cases of Figures 1(a) and (b). Figure 16(c) shows a topological transition in which a surface closes a hole connecting the outer-surface with the internal void. The closing transition yields a parent-child relation between the two disconnected surfaces, the outer-surface and the inner-surface defining the void. Figure 16(d) shows a translucent rendering of the topological transition of Figure 16(c), which makes the inner sphere visible through the outer-surface. Figures 16(e) and (f) reveal two ways to morph a torus into a sphere by attaching a single topological handle H_1. In Figure 16(e), the handle H_1 is pasted to the outer-surface of the torus, while in Figure 16(f), the same handle is pasted to the inner-surface of a torus. A topological handle H_1 could also transform a connected tori into a simple torus, which results in the ”8”-to-”0” morphing sequence shown in Figure 16(g).

These results demonstrate the ability of our methodology to explicitly control the topological transitions between 3D meshes having different topological types.

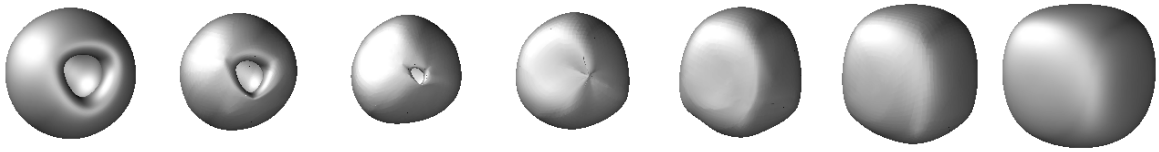
5. Conclusion

We have presented a new method to explicitly control morphing between 3D meshes having different topological types. Our method directly interpolates between the input 3D meshes by using a 4D tetrahedral hypersurface. If the 3D meshes have different surface topological types, the interpolator hypersurface must contain one or more critical point. By pasting a topological handle to relate the 3D meshes around the critical point, our method uniquely specifies the topological evolution of the 3D meshes through the critical point of the 4D hypersurface. We showed that all the possible topological transitions are classified into ten types, and that only four kinds of topological handles are necessary to specify one of these ten types. We have introduced what we call key-frames to implement the effect of the topological handle. A key-frame has two faces, that are, 3D meshes having different topological types, each of which is homeomorphic to one of the input 3D meshes. The

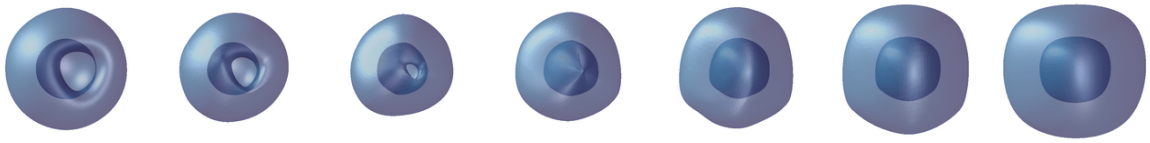
(a) One of the two isosurfaces disappears (cf. Figure 5).



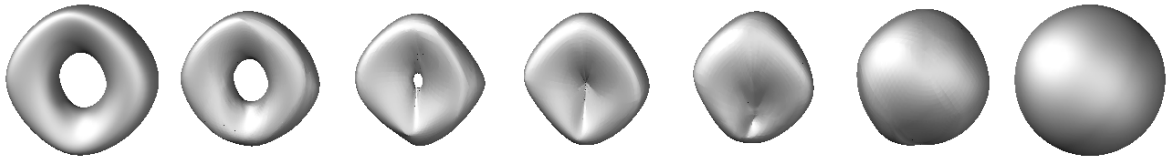
(b) Two disconnected isosurfaces join (cf. Figure 6(a)).



(c) The opening to a void within a solid is closed (cf. Figure 7(b)).



(d) The translucent rendering of (c).



(e) The hole of a torus is plugged (cf. Figure 7(c)).

(f) The ring of a torus is cut (cf. Figure 7(d)).

(g) Morphing between “8” to “0” (cf. Figure 7(d)).

Figure 16. Morphing results.

two faces of the key-frame bridge two surface topologies before and after the topological transition, which is induced by one of the four topological handles.

The formalism presented in this paper allowed us to specify spatial inclusion relationships of time-varying surfaces at any point in time by tracing the sequence of topological transitions. For a non-expert user to take advantage of this powerful formalism, we obviously need a convenient user interface to specify one of the ten possible topological transitions enumerated in Figures 4–7. Note here that the user is on a safe ground since our formalism automatically excludes such topologically unsound events as illegal surface self-intersection. The user interface must also allow us to easily specify necessary features of a key-frame needed to realize the topological transition specified. In addition to the desired topological change, a key-frame must know a point, both in time and space, of topological transition. It must also have an appropriate geometry as well as vertex connectivity so that the meshes before and after the topological transitions are compatible with the key-frame. We need to develop a multiresolution framework suitable for generating a key-frame and corresponding source and target 3D meshes that are compatible with each other both geometrically and topologically.

It is important to incorporate degenerate topological transitions (cf. Section 2) into our framework, which could produce more attractive morphing effects. Handling of open surfaces is an important area of future research for such surfaces are quite common among existing mesh models. Embedding properties other than inclusion, for example, knotting, is also an area of our future research.

Acknowledgments

This research is supported in part by grants from Japan Society of the Promotion of Science (Grand No. 12780185 and No. 12680432), and the Hayao Nakayama Foundation for Science & Technology and Culture.

References

[1] D. DeCarlo and J. Gallier. Topological evoloution of surfaces. In *Graphics Interface '96*, pages 194–203, 1996.

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

[3] A. Gregory, A. State, M. Lin, D. Monocha, and M. Livingston. Feature-based surface decomposition for correspondence and morphing between polyhedra. In *Computer Animation '98*, pages 64–71. IEEE Computer Society Press, 1998.

[4] T. Kanai, H. Suzuki, and F. Kimura. Metamorphosis of arbitrary triangular meshes. *IEEE Computer Graphics and Applications*, 20(2):62–75, 2000.

[5] A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. In *Computer Graphics (Proceedings Siggraph '99)*, pages 343–350, 1999.

[6] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS: Multiresolution Adaptive Parametrization of Surfaces. In *Computer Graphics (Proceedings Siggraph '98)*, pages 95–104, 1998.

[7] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, Jan. 1997.

[8] R. Ohbuchi, Y. Kokojima, and S. Takahashi. Blending Shapes by Using Subdivision Surfaces. *Computers and Graphics*, 25(1):41–58, 2001.

[9] Y. Shinagawa. *A Study of a Surface Construction System Based on Morse Theory and Reeb Graph*. PhD thesis, Department of Information Science, Graduate School of Science, University of Tokyo, 1992.

[10] Y. Shinagawa, Y. L. Kergosien, and T. L. Kunii. Surface coding based on morse theory. *IEEE Computer Graphics and Applications*, 11(5):66–78, Sep. 1991.

[11] S. Takahashi. Multiresolution Constraints for Designing Subdivision Surfaces via Local Smoothing. In *Computer Graphics and Applications (Proc. of Pacific Graphics '99)*, pages 168–178, Los Alamitos, Oct. 1999. IEEE Computer Society Press.

[12] G. Taubin. A signal processing approach to fair surface design. In *Computer Graphics (Proceedings Siggraph '95)*, pages 351–358, 1995.

[13] G. Turk and J. F. O'Brien. Shape transformation using variational implicit functions. In *Computer Graphics (Proceedings Siggraph '99)*, pages 335–342, 1999.