(appeared in *Graphics Interface '96*, pp. 194-203)

Topological Evolution of Surfaces

Douglas DeCarlo and Jean Gallier
Department of Computer & Information Science, University of Pennsylvania, Philadelphia PA 19104-6389
dmd@gradient.cis.upenn.edu, jean@saul.cis.upenn.edu

Abstract

This paper presents a framework for generating smooth-looking transformations between pairs of surfaces that may differ in topology. The user controls the transformation by specifying a sparse *control mesh* on each surface and by associating each face in one control mesh with a corresponding face in the other. The algorithm builds a transformation from this information in two steps. The first step constructs a series of shapes and meshes (according the theory of topological surgery) that describes how topological changes should occur at critical points during the transformation. This makes possible the second step, which establishes smooth transformations by combining intermediate shapes in this series. Control meshes allow the user precise but intuitive control of the morph, while the 3D surfaces that result can be used for rendering or keyframing.

Résumé

Cet article presente une méthode pour engendrer des transformations continues entre deux surfaces dont la topologie peut etre différente. L'utilisateur controle la transformation en spécifiant une *grille de controle* sur chacune des surfaces, et en associant a chaque face de l'une des deux grilles une face correspondante de l'autre. Notre algorithme construit une transformation à partir de cette information en deux étapes. La première étape construit une suite de formes et de grilles (suivant la théorie de la chirurgie topologique) qui decrivent la façon dont les changements de topologie doivent se produire en certains moments critiques. Ceci rend possible la deuxième étape, qui construit une transformation continue en combinant les étapes intermédiaires. Les grilles de controle donnent à l'utilisateur un controle precis mais intuitif de la métamorphose, et les surfaces qui en résultent peuvent être utilisées pour la visualisation ou l'animation.

Keywords: Surface morphing, Surface evolution, Control mesh

1 Introduction

Your local topologist will tell you that you cannot deform an orange into a donut. But what if you want to do this anyway? There is hope, but you'll need more than a simple deformation: you must also evolve the topology, adding a hole where there was none. The theory of topological surgery describes several ways this hole might be added, but the appropriate surgery for a particular transformation is largely an artistic decision.

The task of transforming an orange into a donut epitomizes the general problem of metamorphosis between two objects, commonly called "morphing". Morphing starts from a *correspondence* between the two objects, that specifies where features on one object end up on the other object as a result of the transformation. When the transformation involves topological change, the correspondence must also indicate how the change takes place. The morphing engine effects a transition that realizes the desired correspondence, using a method of *interpolation*.

In recent years, image morphing techniques have gained considerable popularity, especially in the entertainment industry. This success depends in part on algorithms that allow an animator to specify visually engaging correspondences for image morphs in an intuitive way. Unfortunately, any intermediate forms produced by image morphing methods exist *only* in image form. Surface models are often required in animation for keyframing, or to allow shadows or lighting effects to be computed.

Meanwhile, for the metamorphosis of 3D surface models, most research has focused either on morphing between a restricted, topologically similar class of shapes, or on automatically constructing the correspondence between the two shapes for a morph. Often, the user has little or no say in how the morph takes place.

This paper focuses on the specification of correspondence in the presence of topologically different shapes, and the interpolation issues that arise in the presence of topological evolution. In particular, we investigate the use of a sparse *control mesh* to define the transformation. Correspondences between faces of the control mesh induce correspondences between points on the objects; discrepancies between the structure of corresponding faces describe the topological evolution that must occur during the transformation. The main contribution of this paper is that the framework described here allows the smooth transformation between topologically different models while providing the animator with control over the morph.

The organization of the paper is as follows. After an initial introduction of basic concepts and previous work in section 2, the topological and geometric issues involved in evolution are discussed in section 3. A treatment of several implementation questions follows in section 4. After a brief summary in section 5, we conclude after illustrating sample transformations involving the evolution of topology in section 6.

2 Background and Related Work

When concerned with topology of surfaces represented by a mesh, there is some terminology that needs to be defined. The *surface topology* of the shape is specified by the connectivity of the surface. For example, a sphere and torus have different surface topologies. The *mesh topology* is specified by the graph connectivity of the mesh. The *geometry* of a shape is a specification of the locations of the nodes in space. Clearly, if two shapes have different surface topologies, then their mesh topologies must differ. Also, a simple deformation of a shape (a geometric change), does not change either the mesh or surface topology. We will later see how a combination of geometric deformation and topological surgery yields the desired transformation.

The morphing of images, 2D curves and 3D surfaces is an active area of current research. Beier and Neely [1] describe an image morphing technique which allows the user to specify corresponding features between two images using directed line segments. For those regions of the image not covered by line segments, a weighted average of features is used. This sparse specification of features seems to allow the user the most intuitive form of control for morphing. Morphing techniques for 2D curves have also been developed. Addressing the problem of correspondence, Sederberg and Greenwood [15] blend between two 2D polygonal shapes using a correspondence extracted by analogy with the bending and stretching of wire. A multiresolution approach to 2D curve morphing was presented by Goldstein and Gotsman [5].

Addressing 3D models, Kent et al [8] consider the morphing of polyhedral objects topologically equivalent to a sphere. This work primarily concerns the automatic generation of correspondences between shapes, but also includes an algorithm to merge the meshes of two polyhedral shapes. Parent [14] improves this method, using a recursive algorithm to find a correspondence between any two topologically equivalent shapes. Lazarus and Verroust [10] generate a correspondence, while giving the user rough, high-level control by specifying two axes—one in each object. Kaul and Rossignac [7] produce interpolations between shapes by combining scaled versions of the shapes. Hughes [6] performs morphing on volumetrically sampled implicit surfaces, and improves the smoothness of the transformation by scheduling frequencies using Fourier analysis. Wyvill [17] describes other methods for warping implicit surfaces. Lerios et al [11] show how the Beier and Neely image morphing technique can be extended to volume representations. Methods for altering the topology of the surface mesh during transformation have also been presented. By introducing duplicate or degenerate surface mesh elements, Bethel and Uselton [2] can produce a transition between two surface meshes that differ topologically. Delingette et al [3] use a "simplex mesh", and perform basic mesh operations that can alter the shape topology. The interpolation was performed using a physics-based deformation approach, using a method derived from a data fitting process.

What makes our work different from previous work is that we emphasize both user control and smooth transitions between topologically different objects. This emphasis enables results that can equal the dramatic character seen in image morphing.

Beier and Neely [1] allow the user to specify corresponding features in an image, which enables complete control by the user. Morphing between shapes using blobs (metaballs) only allows a very rough correspondence to be established between blobs. The user control in [10] provides only rough control over the morph. None of the previously mentioned 3D shape transformation work seriously address the user control issue (although it certainly could be incorporated into some of the work).

3 Surface Morphing and Topological Evolution

A smooth transformation between topologically different objects requires both *morphing* for the geometric interpolation, and *evolution* of the topology. These two processes are closely linked, and only an appropriate combination of them will yield a smooth transformation. We define a smooth transformation as having the following properties:

1. Over the course of the transformation, no discontinuous jumps in shape are present
2. No undesirable topological changes occur (such as the splitting open of a surface)
3. Intermediate stages should not be overly distorted

In virtually all previous morphing work, the first of these is the paramount goal. The second of these was a concern in [6] and [8]. Hughes noted that coarse volumetric sampling can result in topological features appearing suddenly (such as an instantly appearing hole). Kent et al addressed both the second and third points by noting that they cannot be satisfied without using *both* geometric and topological information during the transformation process. Only [3], [7], [8], [10] and [14] use both geometric and topological information in the morphing process.

To perform the morphing, we are given two surfaces S_1 and S_2, which we will refer to as the source shapes.

For this paper, we will assume that the source shapes are triangulated polygon surfaces, although many of the arguments supplied here will apply to any surface representation. We will also assume that the source shapes are orientable surfaces (non-orientable surfaces include Möbius strips and Klein bottles). In the following sections, without loss of generality, we may refer only to the transformation or correspondence from S_1 to S_2.

3.1 Topological Transformation

The transformation between two topologically different shapes will involve an evolution of topology. This can include adding or removing a hole, or puncturing a closed surface so that it develops a boundary. The process of altering the topology of a shape involves *surgery*—cutting and gluing of the surface; additional *deformation* may be required to maintain reasonable geometry.

Figure 1 illustrates the role of surgery in topological evolution; it shows the shapes where surgery is performed in one transformation from a sphere to a torus. Figure 1(a) shows a sphere with two marked points (perhaps the poles). If we cut through the sphere at these points, and stretch these puncture points into circles, the result is the open tube shown in Figure 1(b). By gluing the two circles together, we can form the torus in (c).

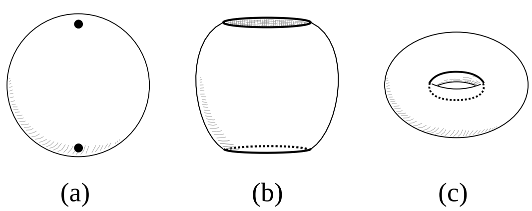


Figure 1: A transformation from a sphere to a torus

An alternative transformation is shown in Figure 2. Starting with the same sphere with two marked points in (a), we can push these points into the sphere until they touch, and then glue them together. The result is the pinched sphere shown in Figure 2(b). From there we obtain the shape in (c) by stretching the pinched point out into a circle that becomes the inner ring of the torus.

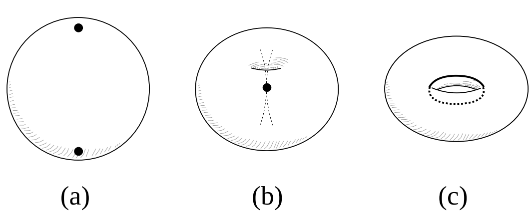


Figure 2: Another transformation from a sphere to a torus

The surgeries of Figure 1 and 2 involve *intermediate shapes*—an open tube or a pinched sphere—which have a different topology from both a sphere and a torus. However, there is an important difference for our purposes. In the transformation of Figure 1, the open tube is realized for an extended period of time while in the transformation in Figure 2, the pinched sphere exists for only a single point in time. In fact, considering the pinched sphere in only geometric terms, we cannot determine whether it is a deformed sphere (formed by pinching) or a torus with the hole closed. So at the moment of surgery, the topology is altered, while the *appearance* of the shape remains unchanged. It is crucial to the smoothness of the transformation that the intermediate shape exists for only a single point in time. Otherwise, we will violate our second condition on smooth transformations.

In order to completely specify a smooth transformation between a sphere and a torus, we must combine the surgery with deformation. A first deformation is required to deform the sphere smoothly into a pinched sphere. After the surgery changes the topology to be that of a torus, another deformation opens the torus hole. In this morphing application, the morphing is responsible for these deformations. As we will see in section 3.4, for certain topology changes, we must place restrictions on the shapes generated by the morphing to ensure the kind of smooth transition shown in Figure 2.

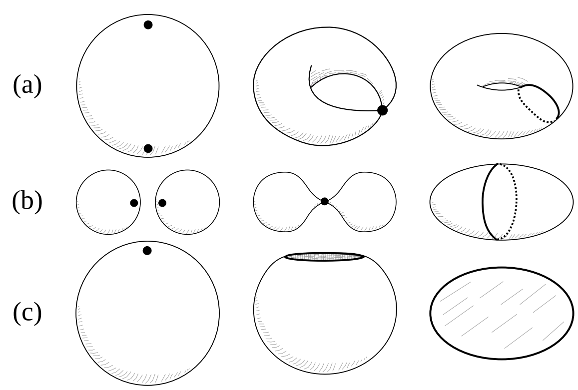


Figure 3: Various transformations

Three additional examples of smooth transformations are shown in Figure 3. The transformation in (a) shows a different type of sphere-torus evolution, sometimes called torus "strangling" [9]. The two points on the sphere are pulled *away* from the surface and glued together on the outside of the sphere. The intermediate shape produced is similar to that of a croissant. The pinched point stretches out to become a ring passing through the torus hole. An example of objects merging or splitting is shown in Figure 3(b). Points on each of the two surfaces are glued together; then this point is stretched into a circle around a single common "blob." Finally, the simple puncture of a sphere in Figure 3(c) shows how it can be unfolded into a flat plate by poking through at a single point (like popping a balloon).

The transformations in Figures 2, 3(a) and 3(b) are very similar. All three transformations have *two* points that are glued together, and then expanded into a circle. In Figure 3(c), a *single* point on the sphere becomes a closed curve (the boundary) of the flat plate. In the description of the control mesh in section 3.2.1, we will see how the correspondence of points (perhaps multiple points), and curves is used to specify the topology changes. The methods presented in this paper handle topological changes that map between one or two points, or one or two circles (closed curves). Open curves may also be used (such as for a "mouth opening"), but will not be discussed here.

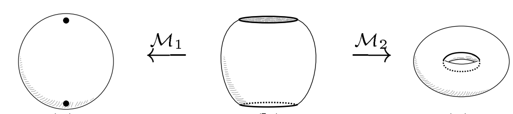
These transformations are qualitatively the same as those used in describing Morse theory [12], which describes the topological changes observed when viewing sweeping cross-sections of surfaces. The concept of gluing is borrowed from topology and the notion of quotient space [13]. We know from the classification theorem for compact surfaces that the operation of gluing is very powerful, since all compact (orientable) surfaces can be obtained by gluing together flat disks. This suggests that the surgery operations described here allow transformations between orientable shapes of arbitrary topology. Koenderink [9] provides numerous examples of "morphological scripts," where he attempts to qualitatively classify shapes based on how they are formed by topological evolution. These scripts can be viewed as recipes for performing evolution, similar in spirit to the examples given above.

3.2 Surface correspondence

The representation of surface correspondence used in previous surface morphing work [8] is a one-to-one mapping between S_1 and S_2. This is a reasonable construction given that the source shapes are topologically equivalent (they are both genus 0 in [8]). But from topology we know that if two surfaces differ topologically (if they are not homeomorphic), then there is no invertible mapping between them. Nevertheless, we must still establish some sort of correspondence between S_1 and S_2.

To address this issue, we construct a new model S, where we can find two surjective (onto) maps: $\mathcal{M}_1$ from S to S_1, and $\mathcal{M}_2$ from S to S_2. Therefore, each node in S has a unique corresponding point in S_1 and in S_2. If S_1 and S_2 are topologically equivalent, then $\mathcal{M}_1$ and $\mathcal{M}_2$ will also be invertible. When S_1 and S_2 differ topologically, then either $\mathcal{M}_1$ or $\mathcal{M}_2$ (or both) will not be injective. In general, the mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ will be locally invertible everywhere except where topology changes occur.

For example, if S_1 is a sphere, and S_2 is a torus, then the constructed model S will have the topology of an open tube, as in Figure 4(b). In a sense, the mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ perform the puncturing and gluing operations associated with the transformation of Figure 1. The mapping $\mathcal{M}_1$ and $\mathcal{M}_2$ will be locally invertible everywhere except on the boundary rings of the tube where topology changes occur. $\mathcal{M}_1$ maps each of the ends of the open tube to its associated pole of the sphere. Basically, the circles on the tube boundaries each collapse to a point. This collapse reflects the fact that the poles are punctured during the transformation. $\mathcal{M}_2$ maps the ends of the tube together into the central ring of the torus. This corresponds to the gluing.



(a) (b) (c)

Figure 4: Mappings from the constructed model S (b) to S_1 (a) and S_2 (c)

In order to build S and the associated mappings, we need to find a merged mesh which contains mesh topology information from both S_1 and S_2. Doing so will allow us to associate nodes in S with locations in S_1 and S_2—information necessary to build $\mathcal{M}_1$ and $\mathcal{M}_2$. For now, we will only be concerned with the locally invertible regions of $\mathcal{M}_1$ and $\mathcal{M}_2$. In section 3.2.3, the regions where topology changes occur will be addressed.

To construct S, we will adapt the technique from Kent et al [8] which produces a common vertex/edge/face network from the source shapes. For each of the nodes in S_1 and S_2, we must add a corresponding node to S. The edges of the shapes are overlaid and also added to S. To perform this algorithm, all that is needed is a correspondence between the two surfaces–for each node in one surface, the corresponding position of the node in the other surface is known.

3.2.1 The Control Mesh

In order to specify the correspondence between S_1 and S_2, the user places a *control mesh* "on top of" each of S_1 and S_2, which we will call C_1 and C_2 respectively. This control mesh is independent of the surface mesh used to define the surface geometry and topology. The nodes of this control mesh are placed on the surface of the shape, and the edges follow the surface of the shape. The exact nature of these surface edges is an implementation issue. The faces of this mesh are generally triangular, with some quadrilateral faces allowed at locations where the topology changes (more details on this later). Example control meshes for a sphere and torus are shown in Figure 5(a) and (b), which produce the transformation shown in Figure 2.

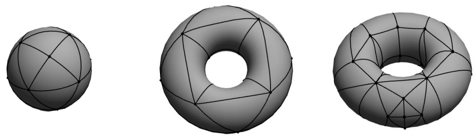


Figure 5: Sample control meshes for a sphere and torus

In addition to defining the control mesh, the user also performs a manual correspondence operation between pairs of faces–taking one from C_1 and one from C_2 (so C_1 and C_2 will have the same number of faces, as well as having the same topology except where surgery occurs). An example of a specified correspondence is shown in Figure 6, with corresponding faces tagged appropriately. By defining this correspondence between two faces, the user is in effect saying that the portion of the surface inside one face will be transformed into the surface contained within the other face. Thus, the specification of the control mesh gives the user complete control over which part of one surface maps to the other surface. Using this control, the user can add to a morph the dramatic character observed in the many animations produced using image morphing. Later, in Figure 13, we will see an example of this precise control.

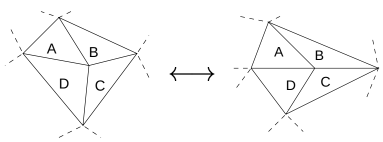


Figure 6: Control mesh correspondence

Given two points that correspond on the models, large regions on one shape around the point on that shape will often map to large regions around the point on the other shape. This coherency information can be used to have a fairly sparse control mesh. In general, there will far fewer faces of the control mesh than in the actual surface mesh describing the shapes. In fact, it is not difficult to have the same control mesh used on identical surfaces triangulated at different resolutions (assuming a correspondence is available between these two triangulated surfaces). This can allow more interactive use of the correspondence system, analogous to the use of lower resolution images in image morphing [1] to speed up the design phase.

The control mesh also allows the user to decide how and where topology changes occur. Gluing of the surface can be accomplished by having the correspondence of two faces that are not adjacent (they share no edges) in one shape with two faces that share an edge in the other shape. Cutting the surface often involves stretching a single point on the surface of the shape into a curve. Where no topology changes should occur, the user constructs a local graph isomorphism between corresponding parts of the control meshes. As seen in Figure 6, the control mesh topologies are the same in the corresponding regions.

For example, the control meshes in Figure 5(a) and (b) correspond to the sphere-torus transformation shown in Figure 2. If we examine the regions near the pole of the sphere, and near the hole of the torus, we will see the correspondence shown in Figure 7. On the left, we see how the four patches that touch a pole of the sphere meet at a single point. This point will be cut, and stretched into the circle at the center of the torus hole, as in Figure 1. The right side of Figure 7 shows the corresponding location of the control mesh on the torus (with the corresponding faces labeled A, B, C and D). Each of the four-sided faces on the torus control mesh have one edge that is collapsed into a point into the sphere mesh. On the other side of the torus central ring, is another group of four-sided faces corresponding to the other pole of the sphere.

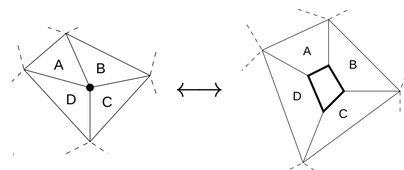


Figure 7: Correspondence involving a topological change

For an analogous reason, using the same control mesh for a sphere (a), and the torus control mesh in Figure 5(c), will result in the transformation in Figure 3(a). The ring of four-sided control mesh faces that collapse into the pinched points on the croissant can be seen near the top of the picture.

3.2.2 Mapping construction

First, consider building the locally invertible portions of the mappings $\mathcal{M}_1$ and $\mathcal{M}_2$. The user specifies a correspondence by giving two faces and a mapping between their edges, as shown in Figure 8(a). Our correspondence must associate a point on S_1 with each point on S_2 and vice versa. Moreover, at face boundaries, points should be mapped to the same surface location—no matter which of the two adjacent control faces determines the mapping. If this does not occur, then points near the boundary can end up overlapping points in adjacent control faces, causing local surface kinks and self-intersections. This restriction is actually fairly simple to attain, if the mapping of points on the boundary is dependent only on information from the two control mesh nodes that are on the common edge.

Given a triangular surface patch specified by a control mesh face, and a point within the patch, we can find a reasonable set of coordinates for this point on this patch

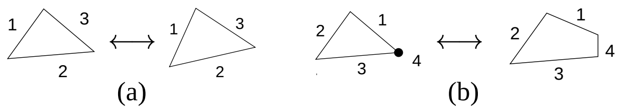


Figure 8: Correspondence specification for control faces

(on the surface). Here, we can use one of the available methods which establish a local coordinate system on the surface of the shape [16]. For the applications here, however, we use a simple "barycentric" map, described in section 4. So given the patch in Figure 9(a) and a point p, we can find its coordinates inside the patch. Also, given the patch in (b), we can find the point q that corresponds to point p.

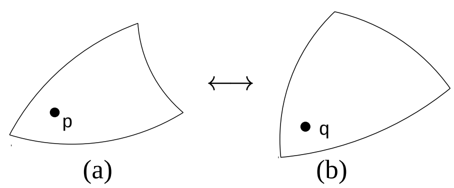


Figure 9: Point-point correspondence

3.2.3 Surface Surgery

Now, consider topology changes, where either $\mathcal{M}_1$ or $\mathcal{M}_2$ is not invertible. Such changes arise when the user associates a four-sided face with a three-sided face, as in Figure 8(b). The key step for such correspondences is to add the additional structure to S needed to define the mappings $\mathcal{M}_1$ and $\mathcal{M}_2$. This additional structure lets $\mathcal{M}_1(S)$ and $\mathcal{M}_2(S)$ overlap on critical features of S_1 and S_2. These overlaps can then be pulled apart during topological evolution.

For example, for the sphere-torus evolution in Figure 1, a correspondence must be found between the two control nodes at the sphere poles, and the inner ring of the torus. For the sphere, we add the nodes in S which correspond to positions of the control mesh nodes on the sphere (we must also subdivide the surface face in S to maintain the triangulation properly). For the torus, we must also add edges (and nodes) along the control edges that specify where the torus will be cut. In Figure 10(a), a mesh is shown with a dotted line indicating the path of the control edge that makes a "cut" along the surface. After cutting, additional nodes are added along this path, as well as additional edges to preserve the triangulation. The result is shown in Figure 10(b). In addition, we *duplicate* these edges and nodes along this control edge. This is because these two parts of the surface will actually separate topologically, since the single ring is evolving into two separate points on the sphere.

Now that this additional structure has been added to S, we can construct the mappings $\mathcal{M}_1$ and $\mathcal{M}_2$. The topology if S is now equivalent to an open tube, as in Figure 1(b). By collapsing each of the tube ends into separate points, we can form the poles of the sphere. By gluing together the tube ends, we can form a torus.

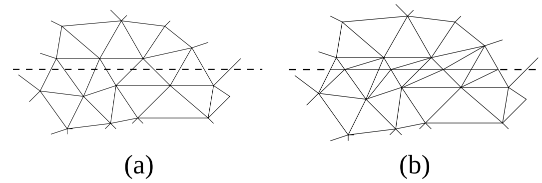


Figure 10: Surface mesh surgery to allow cutting

In general, we must always perform the underlying surface mesh surgery (as in Figure 10) along those control edges (or at those control nodes) where topological changes occur. We must also count the number of connected components in each control mesh that are present at the location of topological change (for the types of topologial changes addressed, there will be either one or two components). The duplication operation described above need only be performed if the number of components differs. For example, the sphere-torus morph has two components in the control mesh of the sphere (the control mesh nodes at the poles), and one component in the control mesh of the torus (a single center ring). Hence, duplication is necessary (as we already know).

For a sphere morphing to a surface with boundary, such as in Figure 3(c), we do not need to perform the duplication of nodes and edges along the boundary. This is because the components of the the control mesh of the sphere (the single control mesh node where the opening occurs) and the control mesh of the open surface (its boundary being the single component) each have a single component. Node identification is all that is necessary to close up the boundary.

So we see that by the addition of appropriate structure to S, we can create a shape that can be easily changed into that of either S_1 and S_2 by simple node identifications. It will also be able to represent any of the intermediate topologies encountered in the morph. During the course of the morph, we can keep track of the current topology of the surface, and perform the appropriate node identifications to produce a shape of the correct topology.

3.3 Interpolation

Once a correspondence is established, a simple linear interpolation of node positions using S will produce a morph from S_1 to S_2. As always, we can parameterize this interpolation using a variable $t \in [0, 1]$, so that when $t = 0$, S has the geometric shape of S_1, and when $t = 1$, S has the geometric shape of S_2.

In addition to linear interpolation, Kent et al [8] suggests Hermite interpolation of nodes, using the surface normals at these nodes as start and end vectors. Alternatively, the control mesh can be used to allow the user to specify different start and end vectors for each control node. For nodes on the surface within control mesh faces, we can interpolate the vectors at the nearby control nodes using the coordinates developed for mapping construction. This additional control can be used to further personalize the morphing process, as well as to help the user avoid surface intersections during the morphing.

We will see in the next section that the motion of nodes as t goes from 0 to 1 must be restricted, so that topology changes occur smoothly.

3.4 Intermediate Geometry

The representation of correspondence developed in section 3.2 is powerful enough to describe the transformation in Figure 1 as well as Figure 2. Additional constraints on transformations are required to avoid abrupt changes in shape and to prevent surfaces from opening. These constraints are needed when the number of connected components in the respective control meshes (described in section 3.2.3) are not equal at locations where topological change will occur. In these cases, direct linear interpolation between the geometries of S_1 and S_2 will produce an open surface.

A simple example will illustrate the problem and motivate our solution. Consider the evolution of a sphere into a torus. If linear interpolation is used, the poles of the sphere will begin to grow into rings as soon as morphing begins, resulting in an open tube. Suppose however that we break the morph into two steps. In the first, we pinch the sphere; in the second, we open the hole. Now the surface is sure to remain closed.

Thus, to keep closed the regions where the surface might split open, we use a series of *intermediate shapes*. These shapes will have "safe" geometries in which any regions which may potentially open up during morphing have been collapsed. Denoting these intermediate shapes as T_1 and T_2 (the safe shapes for S_1 and S_2 respectively), the following sequence of morphs will not open the surface:

$$S_1 \leftrightarrow T_1 \leftrightarrow T_2 \leftrightarrow S_2$$

Performing topology alteration and morphing deformation as separate steps would produce a very restrictive morph. We will soon see how we can perform both at the same time by only moderately restricting the morphing deformation.

To generate an intermediate shape, we must perform a deformation which collapses regions that are ultimately pulled apart during topological evolution. The regions on the shape where this collapsing must occur are specified by control *curves*, which are sequences of control mesh edges formed from the connected component at the location of surgery. For the sphere-torus evolution, the desired control curve are those edges along the center ring of the torus that are identified with the sphere poles. Methods for collapsing control curves that lie on surfaces were presented in [16]. The goal of these methods is to maintain the surface while collapsing a control curve into the point $\mathbf{c}$. For our applications, we use a simple deformation method, which is described in section 4. Figure 11(a) shows the result of applying such a deformation to the torus in Figure 5(b).

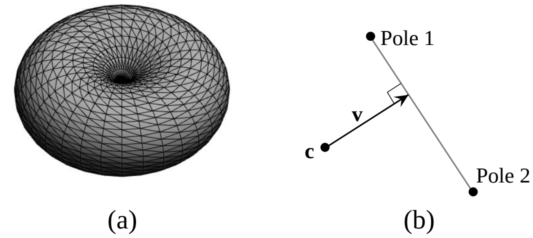


Figure 11: (a) Intermediate shape produced for a torus (b) Restricting the motion to allow smooth transitions

Once this intermediate shape is generated, it can be used to smoothly morph between S_1 and S_2. Suppose we are doing a torus-sphere morph, where S_1 is a torus, and S_2 is a sphere. Then if we have generated T_1, the intermediate shape for S_1, we can create a deforming model $R_1(t_s) = S_1(1-t_s) + T_1 t_s$ by simple node position interpolation (since T_1 is simply a deformed version of S_1). The parameter t_s changes with the morphing parameter t, so that as t changes from 0 to t_{surgery} (the user-specified time that the topology change occurs), t_s changes from 0 to 1. We can then morph between the geometries of R and S_2. After $t = t_{\text{surgery}}$, we can be assured that the morph will proceed reasonably, since the torus hole will have collapsed to a point.

As noted earlier, morphs from S_1 to R to S_2 are undesirable. To improve the quality, we note that we only need to perform this restriction "near" the surgery location—all other nodes can move freely. We can define a distance function $D(\mathbf{p})$ which measures if the point $\mathbf{p}$ is nearby the surgery location as follows:

$$D(\mathbf{p}) = \begin{cases} 0 & \text{if } D_{\text{surg}}(\mathbf{p}) > d_{\max} \\ 1 - \frac{D_{\text{surg}}(\mathbf{p})}{d_{\max}} & \text{otherwise} \end{cases} \qquad (1)$$

where $d_{\max}$ is a user-specified constant controlling the extent of the region, and $D_{\text{surg}}(\mathbf{p})$ is the shortest distance from the point $\mathbf{p}$ to the surgery location measured along the surface. $D(\mathbf{p})$ is 1 for points at the surgery location, and 0 for points far away.

Using D, we can give even more freedom to those nodes near the surgery location, by allowing them to move in a "safe" direction—the direction that does not spatially separate the duplicated nodes. Figure 11(b) shows how to compute the safe direction $\mathbf{v}$ using $\mathbf{c}$, and the location of the poles from the control mesh C_2. The idea is as follows. First, ordinary interpolation is performed. Then, the displacement vector between the current position and that in R_1 is computed. The tangential component of this displacement in the direction of $\mathbf{v}$ is then computed, and is cancelled by a factor of D. Hence, if a node is on the control curve, this tangential component is completely cancelled. This effect falls off as one moves away from the curve. This method allows the surgery locations to move, even during the topological change.

4 Implementation

Figure 12 shows the layout of the user interface used to produce the animations (on an SGI platform). The interface is similar to those used in image morphing, with side-by-side object views, and seems intuitive.

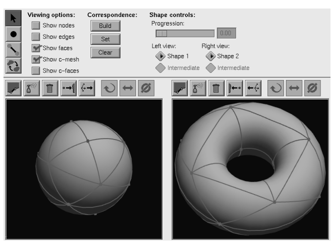


Figure 12: User interface for specifying control mesh

Section 3.2.2 described the need for a method for generating coordinates within a surface patch—we use a simple projection method. This places a restriction on the size and shape of the faces of the control mesh: the surface patches specified by the user must be fairly flat, so the projection is unique.

Given the control mesh face correspondences in Figure 8, we need to perform two operations:

- Given a control mesh face and a point p, find the barycentric coordinates λ of p within the face
- Given a control mesh face and barycentric coordinates λ, find the point p on the surface which has these barycentric coordinates

By projecting upward from the surface onto a plane, we can determine barycentric coordinates for points on the shape surface. These barycentric coordinates are unique, and also adhere to the boundary restriction mentioned in section 3.2.2.

Section 3.4 describes the need for intermediate shapes to ensure the smoothness of the transition. These intermediate shapes have control curves which have been collapsed into a point $\mathbf{c}$. We can find $\mathbf{c}$ as the evenly weighted barycenter of the curve. A deformation which maps all points on this curve to $\mathbf{c}$ is constructed. Nearby points are also moved toward $\mathbf{c}$ to provide a smooth deformation, and D is used to restrict its effect as distance from the control curve increases.

5 Summary

Here, we briefly summarize the steps necessary to perform the evolution of shape given the two source polygonal shapes. The user specifies control meshes on top of each of the source shapes and specifies correspondences between the faces of the control meshes. The user also specifies times for each topological change. The computer does the following:

- Computes correspondence information (described in section 3.2) from the control meshes in the locations where surgery does not occur.
- Creates a unified mesh, S, by combining the meshes of the two source shapes [8].
- Adds additional edges along control mesh edges (as in Figure 10) that permit the shape to "split" along these boundaries (see section 3.2.3). After these "splitting" nodes have been added, the remainder of the correspondence information is computed.
- Computes the intermediate shapes for the transformation (see section 3.4).
- For any time $t \in [0, 1]$:
  - Determines active topological changes
  - Identifies the appropriate nodes of S together, based on the current topology.
  - Interpolates the shapes, restricting the motions of those nodes near surgery locations.

6 Results

Each of the following polygonal surfaces was rendered using SGI OpenGL. Because of this method, the surfaces occasionally show some Mach banding artifacts. These objects are texture mapped, with the texture coordinates interpolated as suggested by Kent et al [8]. Animations described in this section are available at `http://www.cis.upenn.edu/~dmd/evol.html`.

Figure 13 shows how the control mesh can be locally altered to vary the correspondence using a transition from a banana (a) to an orange (c). The three intermediate shapes in Figure 13(b) are produced by varying the control mesh on the orange (in the area that corresponds to the stem of the banana). The leftmost shape is from a transition that people we asked usually preferred. In the center shape, the stem region is larger than the preferred one; in the right one, it is much smaller.

Figure 14 shows two different sequences of a sphere-torus morph. The transition in (a) corresponds to Figure 2, and (b) to Figure 3(a). Of interest to topology fans, is the fact that these two transitions are qualitatively different. In both cases, the poles of the sphere are brought together. Yet in (a), the lines pass through the torus hole, while in (b) they go around the torus hole.

Figure 15 shows morphing from a banana (a) to an open surface with two holes (e). The three topology changes necessary occur at different times. In (b), the banana opens up at the tip to form the border of the surface. The central hole pinches in (c), and opens further in (d) where the ends of the banana have come together. This continues until the resulting final shape in (e). The construction of the control mesh for this example took about 45 minutes, with another 30 minutes of control mesh "tweaking" to get the desired look.

Figure 16 show samples from "Mutafruit", which is a standard dynamics based animation using shapes generated by surface evolution. The morphing violates energy conservation, and also creates interesting inertial forces, creating a lively, almost surreal effect commonly associated with image morphing. In each of these examples, the lighting and shadows present would be extremely difficult and time consuming to produce by image morphing techniques.

7 Conclusion

The main contribution of this paper is that the framework described here allows smooth transformations between topologically different models while providing the animator with control over the morph. Previous systems have not dealt with topology seriously, and many other systems do not provide user control. The presented examples show the new flexibility in surface transformation.

This technique could be extended in a number of ways. The generation of the control meshes can become rather tedious for large objects. Most of the repetitive effort could be eliminated by using a multiresolution representation of the underlying mesh [4] in which large regions of the control mesh would duplicate the underlying mesh—at a lower level of detail. Combining this method with existing correspondence generators [8, 14] would also prove useful. Nevertheless, it is unreasonable to expect an aesthetically pleasing morph (either surface or image) without a detailed user specification.

This system could also benefit from more powerful facilities for avoiding self-intersections during morphs. In our system, the specification of initial directions at the control nodes can help the user avoid some intersections. A more flexible strategy would be to allow the user to specify additional topology changes not vital to the transformation. For example, in a transformation from a torus to a knotted torus (which are topologically equivalent), the user could avoid self-intersection by cutting the torus (an extra topological change) and then tying the knot. In general, self-intersections can cause problems for all surface morphing systems, and finding general techniques remains an important research issue.

Finally, the extension of this method to spline surfaces and time dependent shapes (perhaps articulated) will produce even better animations.

Acknowledgments

We would like to thank Herman Gluck for his helpful discussions. This research is partially supported by NSF grant IRI-9309917 and ONR grant NOOO14-93-1-1217.

References

[1] T. Beier and S. Neely. Feature-based image metamorphosis. In *Proceedings SIGGRAPH '92*, volume 26, pages 35–42, July 1992.

[2] E. Bethel and Samuel P. Uselton. Shape distortion in computer-assisted keyframe animation. In N. Magnenat-Thalmann and D. Thalmann, editors, *State-of-the-art in Computer Animation: Proceedings of Computer Animation '89*, New York, 1989. Springer-Verlag.

[3] Hervé Delingette, Yasuhiko Watanabe, and Yasuhito Suenaga. Simplex based animation. In N. M. Thalmann and D. Thalmann, editors, *Models and Techniques in Computer Animation*, pages 13–28. Springer-Verlag, 1993.

[4] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings SIGGRAPH '95*, pages 173–182, 1995.

[5] E. Goldstein and C. Gotsman. Polygon morphing using a multiresolution representation. In *Graphics Interface '95*, pages 247–254, 1995.

[6] J. F. Hughes. Scheduled Fourier volume morphing. In *Proceedings SIGGRAPH '92*, volume 26, pages 43–46, July 1992.

[7] A. Kaul and J. Rossignac. Solid-interpolating deformations: Construction and animation of PIPs. *Comput. & Graphics*, 16(1):107–116, 1992.

[8] J. R. Kent, W. E. Carlson, and R. E. Parent. Shape transformation for polyhedral objects. In *Proceedings SIGGRAPH '92*, volume 26, pages 47–54, July 1992.

[9] J. J. Koenderink. *Solid Shape*. MIT Press, 1990.

[10] F. Lazarus and A. Verroust. Feature-based shape transformation for polyhedral objects. In *Fifth Eurographics Workshop on Animation and Simulation*, 1994.

[11] A. Lerios, C. Garfinkle, and M. Levoy. Feature-based volume metamorphosis. In *Proceedings SIGGRAPH '95*, pages 449–456, 1995.

[12] J. Milnor. *Morse Theory*. Princeton University Press, 1969.

[13] J. W. Milnor. *Topology from the Differentiable Viewpoint*. University Press of Virginia, 1965.

[14] R. E. Parent. Shape transformation by boundary representation interpolation: a recursive approach to establishing face correspondences. In *Journal of Visualization and Computer Animation*, volume 3, pages 219–239, 1992.

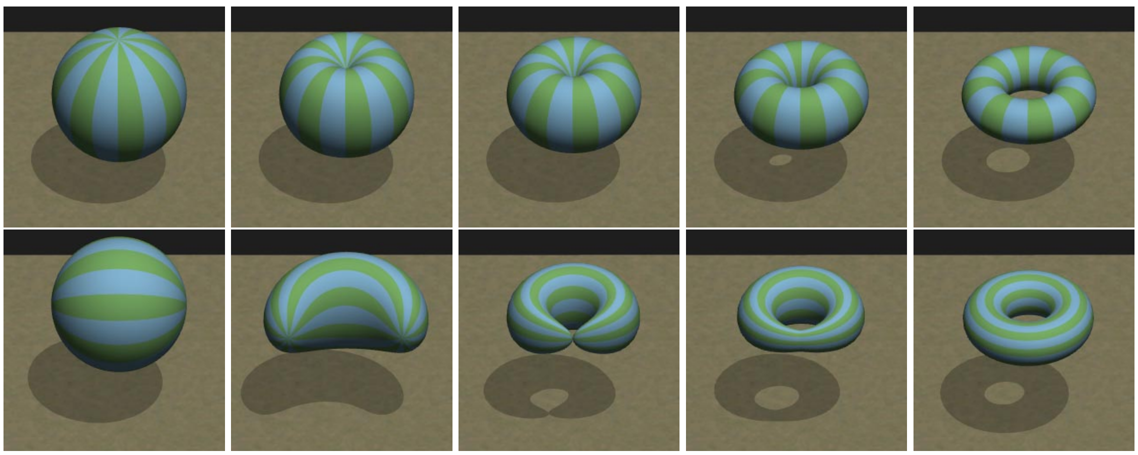
[15] T. W. Sederberg and E. Greenwood. A physically based approach to 2D shape blending. In *Proceedings SIGGRAPH '92*, volume 26, pages 25–34, July 1992.

[16] W. Welch and A. Witkin. Free–Form shape design using triangulated surfaces. In *Proceedings SIGGRAPH '94*, volume 28, pages 247–256, July 1994.

[17] B. Wyvill. Metamorphosis of implicit surfaces. In *SIGGRAPH '90 Course 23: Modeling and Animating with Implicit Surfaces*, 1990.

(a) (b) (c)

Figure 13: Example of correspondence control (banana stem size variation)



(a)

(b)

Figure 14: Morphing between a sphere and a torus (a) pinched sphere (b) strangled torus

(a) (b) (c) (d) (e)

Figure 15: Morphing between a banana and a genus 2 bordered surface

(a) (b)

Figure 16: From “Mutafruit”: (a) morphing from an apple to a banana (b) morphing from a mushroom to a donut