Interactive and User-Friendly Construction of Multi-Segment Curved Handles

VINOD SRINIVASAN
Visualization Science Program
Texas A&M University

ERGUN AKLEMAN [*]
Visualization Science Program
Texas A&M University

JIANER CHEN
Department of Computer Science
Texas A&M University

Abstract

In this paper, we present a method to interactively create multi-segment, curved handles between two star-shaped faces of an orientable 2-manifold mesh or to connect two 2-manifold meshes along such faces. The presented algorithm combines a very simple 2D morping algorithm with a Hermite interpolation to construct the handle. Based on the method, we have developed a user interface tool that allows users to simply and easily create multi-segment curved handles.

1 Introduction and Motivation

This paper presents a new modeling approach for interactively creating very high genus 2-manifold smooth meshes, based on creation of free-form handles[1]. The method can be used for handle creation (i.e., adding a handle to a surface) and for surface blending (i.e. connecting two distinct surfaces). Both applications of the algorithm are useful to designers for creating manifolds of high genus.

A handle is not an extrusion (or lofting) and handle creation is not simply an extrusion method. Handle creation is a topological operation and it requires topological consistency. Therefore, handle creation methods are different than extrusion methods since they are required to guarantee topological consistency. Our method not only guarantees 2-manifold property of final mesh, in every stage of our handle creation costructed meshes continue to be 2-manifold.

The meshes constructed by our method can be smoothed by any subdivision scheme since the manifold property is preserved after handle creation. Figure 1 shows two views of a genus-6 shape created by our approach combined by subdivision schemes. To create this object, we first created 6 multi-segment curved handles by connecting pairs of neighboring faces of a dodecahedron, and then smoothed the resulting mesh using one iteration of the Doo-Sabin and two iterations of the Catmull-Clark subdivision schemes [9, 11, 23].

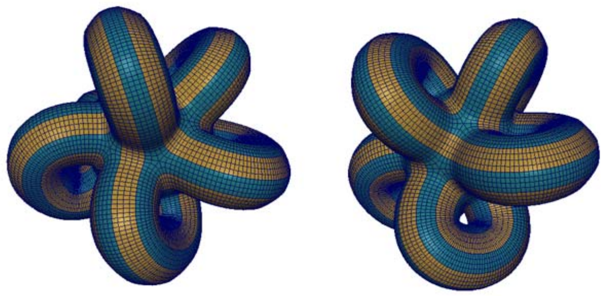


Figure 1. Two views of a genus-6 manifold mesh created by our approach.

Handle creation can also be applied after a subdivision operation, since most subdivision schemes also preserve the manifold property. As a result, subdivision schemes and handle creation operators can alternatively be applied to hierarchically construct high genus manifold meshes. An example of such a hierarchically constructed mesh is shown in Figure 2.

A prototype system for creating multi-segment curved handles was tested in a graduate level shape modeling course in which majority of the students had an architecture

[*]Corresponding Author: Address: 216 Langford Center, College Station, Texas 77843-3137. email: ergun@viz.tamu.edu. phone: +(409) 845-6599.

[1]A two-page summary version of this paper (that does not include methods and algorithms) is accepted by Pacific Graphics'02 as a poster paper.

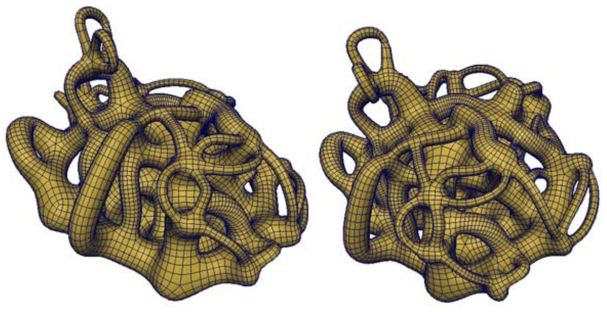


Figure 2. Two views of a manifold mesh created by alternatively creating handles and applying Catmull-Clark subdivision scheme.

undergraduate background. An example of a shape created by the students is the high genus "Möbius band looking" shape shown in Figure 3 The student learned how to use the software and finished the model in one week. For more examples see this year's course webpage at http://www-viz.tamu.edu/courses/viza657/02fall/homework/hw03.html.

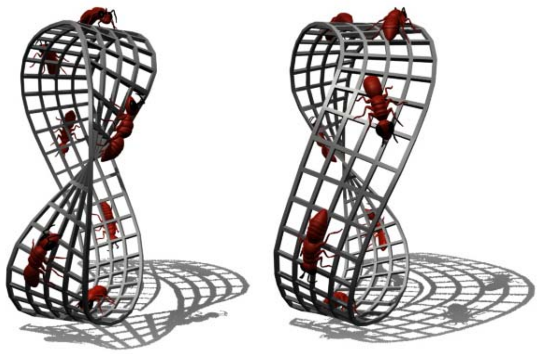


Figure 3. An homage to Escher: *Band Van Möbius II* **[12] by Avneet Kaur. (Ant model was created earlier.)**

The creation of very high genus smooth 2-manifold surfaces has always been a research interest in computer graphics and shape modeling [14]. Ferguson, Rockwood and Cox used Bézier patches to create high genus 2-manifold surfaces [13]. Welch and Witkins used handles to design triangulated free-form surfaces [20]. Using morse operators and Reeb graphs, Takahashi, Shinagawa and Kunii developed a feature-based approach to create smooth 2-manifold surfaces with high genus [19].

We recently introduced a topologically robust mesh modeling approach to computer graphics and shape modeling [1] by adopting topological graph theory [10, 15], a relatively obscure mathematical theory that was introduced almost 100 years ago and known by only a handful of graph topologists. We have shown that this method and subdivision schemes reinforce each other [2, 3]. Our 2-manifold mesh modeling scheme is based on a minimal set of manifold preserving operators [1] that is simple, intuitive and user-friendly.

We have also shown that these operators can be efficiently implemented on almost every mesh data structure including winged-edge and half-edge [5]. Based on this minimal operator set, a user interface is developed and user friendly high level operators to interactively model orientable 2-manifold meshes are introduced [4]. One of the most useful high-level operators is the CREATEPIPE operator [4]. Let $\mathcal{M}$ denote a 2-manifold mesh and let v, f and e denote a vertex, a face and an edge respectively. We define a *corner* of the mesh to be a vertex-face pair, $c = (v, f)$ if f contains the vertex v. The CREATEPIPE(c_1, c_2) operator connects two faces which contain the corners c_1 and c_2 such that there is an edge between c_1 and c_2 and there is an edge between other matching corners (in traversal order starting from c_1 and c_2) of the two faces, as shown in Figure 4. In effect this operation creates a pipe made up of only one segment whose ends are defined by the original faces.

The main problem with the CREATEPIPE operator is that the length of each edge in a handle can be much longer than other edges in the mesh $\mathcal{M}$, as shown in Figure 5B. In this paper, we present preliminary results which demonstrate a solution to this problem. Instead of one pipe segment to connect the two faces, we use a pipe with multiple segments. This pipe approximates a curve that starts from the centroid of the first face, in the direction of the face normal with a given magnitude and ends at the centroid of the second face in the opposite direction of the face normal with another given magnitude. The quality of the handle improves considerably with the new approach, as shown in Figures 5C and 5D.

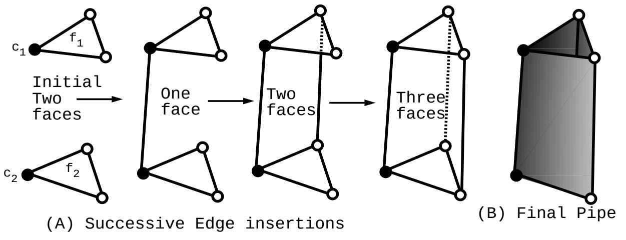


Figure 4. Creation of a pipe by inserting a set of edges.

The remainder of this paper is organized as follows. In Section 2, we introduce a minimal set of operators to im-

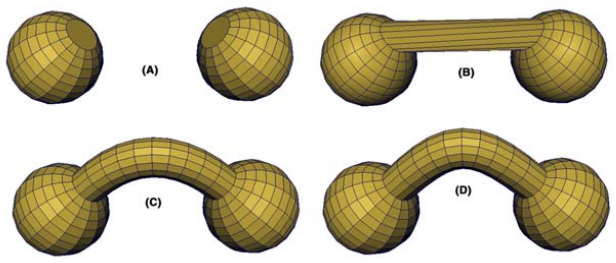


Figure 5. Problem of long edges and our piecewise straight solution. A is the original mesh, B was obtained using the CREATEPIPE **operator, C and D were obtained using our new approach with different weights for the normals in the Hermitian curve equation.**

plement an algorithm to create handles with multiple segments. Section 3 presents an algorithm to create multi-segment pipes. In Section 4, we describe the implementation of the algorithm presented in Section 3. Section 5 shows the role of subdivision and gives several examples of shapes created using our method and discusses the results obtained. Finally, our conclusions are given in Section 6.

2 Minimal Operators

In this section, we describe the operators that are used to implement the handle creation algorithm. These operators can be implemented in any data structure [5] and allow us to change the topology of a 2-manifold mesh [10, 1, 2]. These operators are topologically robust; therefore our algorithm always guarantees topological consistency of 2-manifold meshes. Two fundamental topology change operations and two high level operations are sufficient to develop the algorithm.

1. CREATEVERTEX(v) creates a 2-manifold surface with one vertex v and one face f which we call a *point sphere*. The operation is the same as the Euler operation $MVFS$ [16] and effectively adds a new surface component to the current 2-manifold. The CREATEVERTEX operator is essential in the initial stage of the creation of a new mesh and creates a new surface component in the given 2-manifold. In particular, it is necessary when a new surface component is created in an empty manifold.

2. INSERTEDGE(c_1, c_2, e) inserts a new edge e into the mesh structure between two corners c_1 and c_2.

   In general, if INSERTEDGE inserts an edge between two corners of the same face, the new edge divides the face into two faces without changing topology. On the other hand, if INSERTEDGE inserts an edge between corners of two different faces (this includes the situation in which an endpoint or both endpoints of the new edge correspond to point-spheres), the new edge merges the two faces into one and changes the topology of the 2-manifold.

For the development of the handle creation algorithm we also use two high-level operators that simplify the steps of algorithm. Both of these can be implemented efficiently using the two fundamental operators given above.

3. CREATEFACEMANIFOLD$(v_0, v_1, \dots, v_{N-1})$. creates a two sided face (a manifold surface). This operation can be implemented by the following procedure.

```
(a) for $i = 0$ to $N-1$ do
        CREATEVERTEX$(v_i)$;
(b) for $i = 0$ to $N-1$ do
        INSERTEDGE$(v_i, v_{(i+1) \pmod N}, e)$.
```

   Remark 1. Each of these vertices before the insert edge operation is valence 1. Therefore, we do not have to specify the corners.

4. CREATEPIPE(c_1, c_2) This operator is the same as the one described in the introduction. It connects the two faces which contain the corners c_1 and c_2 such that there is an edge between c_1 and c_2 and there is an edge between other matching corners (with reference to c_1 and c_2) of the two faces as shown in Figure 4.

   Implementation of this operation based on the fundamental operations is also straightforward: instead of a single call to the fundamental operator INSERTEDGE that inserts a single new edge, we make multiple calls to INSERTEDGE to insert edges between all the corresponding corners of the two faces.

   Remark 2. For the implementation of this operator, the two faces do not have to have the same number of corners [4].

An algorithm based on these operators is described in the next section.

3 Creating Multi-Segment Pipes

Our algorithm to create multi-segment pipes, shown in Figure 6, proceeds as follows for a given mesh $\mathcal{M}$ and corners $c_1 = (v_1, f_1)$ and $c_2 = (v_2, f_2)$:

- **Step 1:** for $i = 1$ to $k-1$ where k is the number of segments required in the handle
  3.1. Compute the positions of the vertices in face $f'_i =$

$(v'_{i,0}, v'_{i,1}, \ldots, v'_{i,N-1})$ based on the original faces f_1 and f_2.

3.2. CREATEFACEMANIFOLD$(v'_{i,0}, v'_{i,1}, \ldots, v'_{i,N-1})$.

Remark 3. This step creates a set of manifold faces starting from f'_1 and ending at f'_{k-1} as shown in Figure 6B.

Remark 4. Note that for the sake of simplicity this explanation implies that f_1 and f_2 have the same number of vertices. We will later show that f_1 and f_2 do not have to have the same number of vertices for the implementation of the algorithm.

- **Step 2:** Starting from $f'_0 = f_1$ and ending at $f'_k = f_2$, going through the sequence of faces $f'_0, f'_1, \ldots f'_{k-1}, f'_k$ created above, apply the CREATEPIPE operator to each pair of adjacent faces whose normals point towards each other, using the matching corners as defined by the face traversal starting from c_1 and c_2 (see Figures 6C-6F).

Remark 5. Each face created in step 3.2 is actually two new faces, which share the same vertices and edges, but point in opposite directions. When applying the CREATEPIPE operator in Step 2, the corners should be chosen such that they belong to faces that are adjacent to each other (w.r.t. the sequence of faces created, but excluding faces which share the same vertices and edges) and point in opposite directions.

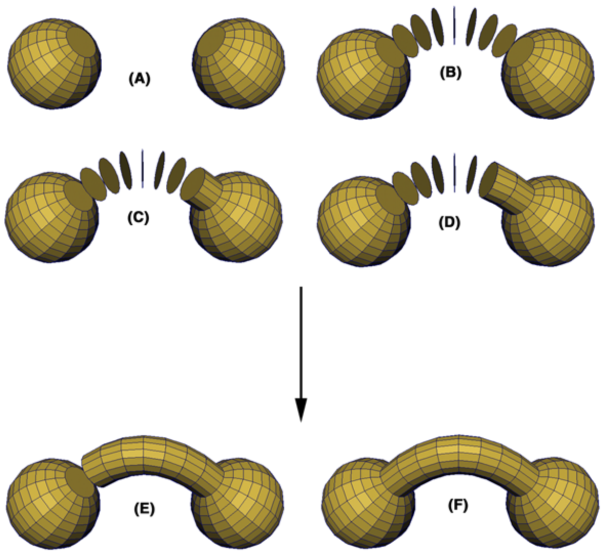


Figure 6. Algorithm.

3.1 Curved Handles

This subsection presents our method to create curved handles between two distinct faces of a mesh. This method is used to compute the positions of the vertices in the intermediate faces (f'_i) described in the previous subsection.

Let two corners $c_1 = (v_1, f_1)$ and $c_2 = (v_2, f_2)$ and two weights w_1 and w_2 be given. If f_1 and f_2 are star shaped faces with the same number of vertices, our goal is to create a multi-segment pipe between the two faces f_1 and f_2 such that the two vertices v_1 and v_2 are directly connected by a series of straight line segments that approximate a Hermitian curve

$$\mathcal{H}(t) = p_1 h_1(t) + p_2 h_2(t) + w_1 n_1 h_3(t) - w_2 n_2 h_4(t). \quad (1)$$

Here p_1 and p_2 are the centroids and n_1 and n_2 are the average normals of f_1 and f_2 respectively (with $|n_1| = 1$ and $|n_2| = 1$) and $h_1(t), h_2(t), h_3(t)$ and $h_4(t)$ are the Hermitian basis functions. This Hermitian curve starts from the centroid of face f_1 in the direction of the face normal (n_1) and ends at the centroid of face f_2, in the opposite direction of the face normal $(-n_2)$. Note that the weights w_1 and w_2 can be negative, which can be used to create holes instead of handles.

3.2 Face Morphing

The Hermitian curve $\mathcal{H}(t)$ determines the overall shape of the pipe. To compute the actual positions of the vertices in the intermediate faces, we need to determine the shape of the cross-section of the pipe at each segment as it goes from f_1 to f_2.

This problem strongly resembles the problem of 2D shape blending [17] (more widely known as 2D morphing). Therefore, our approach is to first simplify the problem into a 2D problem. To morph the shape of the faces from f_1 to f_2 in 2D, we first determine a reference plane $\mathcal{R}$ and rotate both faces to that reference plane. The choice of $\mathcal{R}$ is arbitrary and does not by itself affect the algorithm. We use the vector from the centroid of the first face to the centroid of the second face to determine the unit normal vector $n_{\mathcal{R}}$ to the reference plane $\mathcal{R}$. We rotate f_1 onto $\mathcal{R}$ around its centroid using the rotation axis $n_1 \times n_{\mathcal{R}}$ We rotate the face f_2 similarly. After these rotations, we move each vertex of f_1 and f_2 to the reference plane $\mathcal{R}$, to ensure that the transformed f_1 and f_2 are planar. Both f_1 and f_2 are then translated to make their centroids coincide with the origin of the coordinate system.

3.2.1 2D Morphing

After applying the above operations, the two faces will become coplanar and both of their centroids will be at the same location. Under the assumption that these faces are star shaped and centroid is a star center, the morphing problem is simpler than the general 2D shape blending problem [17, 18]. However, we still need to be careful to avoid self intersections. For instance, if we perform a linear interpolation directly between the vertices of the two faces, we can get self intersections.

Instead of linear interpolation between the vertex coordinates directly, we perform the interpolation in polar coordinates. This interpolation guarantees that intermediate faces do not self-intersect under the assumption that the transformed f_1 and f_2 are star shaped polygons and their centroids are star centers (i.e. any ray enamating from centroid intersect with the face no more than once). Using a reference axis system on the plane $\mathcal{R}$, we resolve every vertex v of f_1 and f_2 into a distance-angle pair (r, θ), where r is the distance of v from the centroid of the face (which is now the origin) and θ is the angle that v (treated as a vector) makes with the X axis. The choice of the reference axis system is arbitrary and has no influence on the final results. We define the reference axis system as follows. We take the vector from the origin to the mid-point of the last edge in f_1 to be the X axis. The Y axis will then be the cross product between $n_{\mathcal{R}}$ and the X axis.

We ensure that the angles always increase monotonically within the same face going from the first vertex to the last one. This is necessary to avoid self intersections as we transition from f_1 to f_2, and also gives a smooth transition. We also ensure that the difference in angles for the first vertices is not more than 180 degrees. This prevents unwanted twists in the pipe (along the axis of the pipe).

Let

$$\begin{aligned} f_1 &= (v_{1,0}, v_{1,1}, \ldots, v_{1,N-1}) \\ f_2 &= (v_{2,0}, v_{2,1}, \ldots, v_{2,N-1}) \\ v_{1,j} &= (r_{1,j}, \theta_{1,j}),\ j = 0 \quad \text{to} \quad N-1 \\ v_{2,j} &= (r_{2,j}, \theta_{2,j}),\ j = 0 \quad \text{to} \quad N-1 \end{aligned} \tag{2}$$

be the resolved representations of the two faces f_1 and f_2, where N is the number of vertices in each face.

Remark 6. To simplify the notation, here we assume that f_1 and f_2 have the same number of vertices. However, this is not a restriction. If f_1 and f_2 do not have the same number of vertices, we simply bisect the edges of the face that has less vertices. A short explanation of the procedure can be given as follows. Let N and M be the number of vertices of f_1 and f_2 respectively. Without loss of generality, let us assume that $N > M$. Then, the number of vertices of f_2 can easily be increased to N by inserting new vertices. After sucha procedure, f_1 and f_2 will have the same number of vertices. Two examples of creating handles between two faces with different numbers of vertices are shown in Figure 7.

We then perform a linear interpolation of the distance-angle pairs using the same parameter t as used for the Hermitian curve, such that $t = 0$ corresponds to f_1 and $t = 1$ corresponds to f_2. Let

$$f'_i = (v'_{i,0}, v'_{i,1}, \ldots, v'_{i,N-1}) \tag{3}$$

be an interpolated face corresponding to a parameter t, where

$$\begin{aligned} v'_{i,j} &= (r'_{i,j}, \theta'_{i,j}),\ j = 0 \quad \text{to} \quad N-1 \\ r'_{i,j} &= (1-t)r_{1,j} + tr_{1,j} \\ \theta'_{i,j} &= (1-t)\theta_{1,j} + t\theta_{2,j} \end{aligned} \tag{4}$$

Using the Hermitian curve $\mathcal{H}(t)$ we get a point $p(t)$ and a tangent vector $n(t)$ for a given parameter t. We rotate $f(t)$ so that its normal points in the same direction as $n(t)$ and translate it so that its centroid coincides with $p(t)$.

We now have a sequence of faces defining the ends of the segments which make up the handle. We finish the process by connecting the corresponding vertices between adjacent faces to create the handle. Three examples of face morphing for different choices of corners are shown in Figure 8. Although, this morphing algorithm cannot guarantee non-self intersection for non-star faces, it can still be used in some applications that include non-star faces. Some examples of such applications are shown in Figure 9. Note that in the examples in Figure 9 handle *"do not self-intersect"* itself although the initial faces are not star shaped.

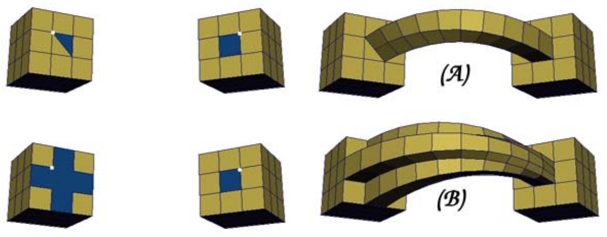


Figure 7. Two examples of creating handles between two faces with different number of vertices. Corners are indicated as white dots and faces to be connected have a darker shade.

Remark 7. To create segments of a handle, we sample the Hermitian curve simply using $t = i/k$. To improve the quality of the handles we tested alternative sampling strategies such as using an appropriate step length with respect to the curvature of the Hermitian curve. However, we concluded that there is no need for any elaborate scheme for interactive handle construction. The main problem occurs when handle intersects itself. However, if the Hermitian curve intersects with itself for the user selected weight values w_1 and w_2, there is no solution to this problem. Since both high curvature and curve self intersection occur for

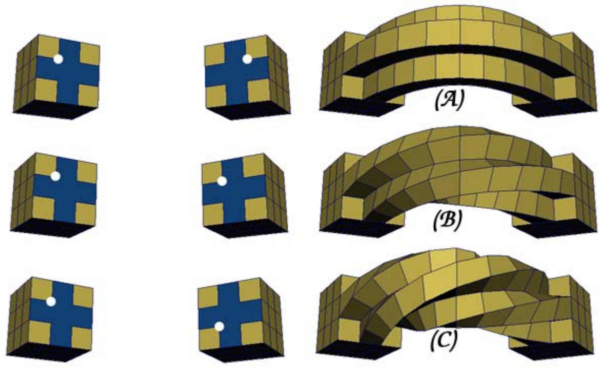


Figure 8. Examples that shows face morphing for different choice of corners. Corners are indicated as white dots and faces to be connected have a darker shade. Note that it is possible to control rotation simply by choosing different corners.

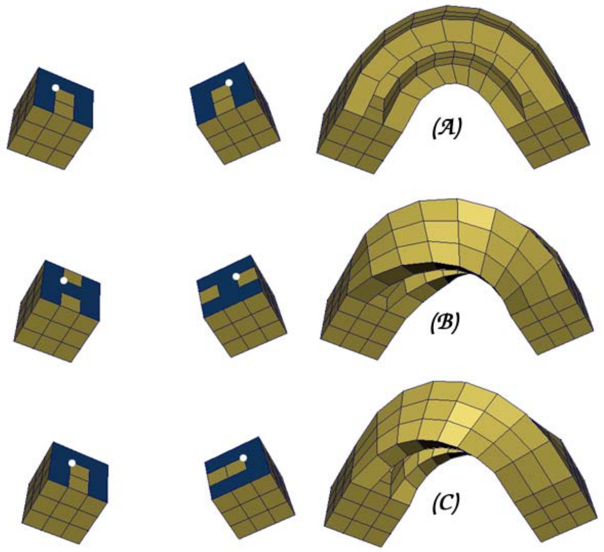


Figure 9. Examples that shows face morphings between two non-star shaped faces. Corners are indicated as white dots and faces to be connected have a darker shade. Note that it is possible to control rotation simply by choosing different corners.

larger values of w_1 and w_2, it is easier to reduce the values of these parameters. Moreover, a very small number of segments is usually sufficient to construct handles. If the curvature is not extremely high, a small number of segments do not intersect with other. Therefore, in our interactive applications users easily avoid self intersection by either decreasing the values of w_1 and w_2 or using less segments.

4 Implementation

The following outlines the implementation of the algorithm to create mult-segment curved pipes, using the methods outlined in the previous sections.

- **Step 1:**
  1.1. If the number of corners of f_1 and f_2 are not equal, until f_1 and f_2 have the same number of vertices, insert new vertices to the face with less vertices.

- **Step 2:**
  2.2. Reverse face f_2.
  2.3. Using the centroids p_1 and p_2 and weighted normals n_1 and n_2 of the two faces, compute a Hermitian curve $\mathcal{H}$

- **Step 3:**
  3.1. Compute the reference plane $\mathcal{R}$ with normal $n_{\mathcal{R}}$.
  3.2. Transform f_1 and f_2 so that their normals are in the same direction as $n_{\mathcal{R}}$ and their centroids are at the origin.
  3.3. Resolve f_1 and f_2 into distance-angle pairs (Equation 2).

- **Step 4:** for $i = 1$ to $k - 1$ where k is the number of segments required in the handle
  4.1. Compute $t = i/k$
  4.2. Compute a point p_i and tangent n_i on the curve $\mathcal{H}$ for t
  4.3. Perform a linear interpolation between the resolved representations of f_1 and f_2 using t as the parameter to obtain the resolved representation of f'_i (Equations 3 and 4)
  4.4. Compute the positions of the vertices in face $f'_i = (v'_{i,0}, v'_{i,1}, \ldots, v'_{i,N-1})$ corresponding to the resolved representation calculated in the previous step.
  4.5. Transform the vertex coordinates of f'_i computed above so that the normal to f'_i points in the direction of n_i and its centroid coincides with p_i.
  4.6. CREATEFACEMANIFOLD$(v'_{i,0}, v'_{i,1}, \ldots, v'_{i,N-1})$

- **Step 5:** Starting from $f'_0 = f_1$ and ending at $f'_k = f_2$, going through the sequence of faces created above, apply the CREATEPIPE operator to each pair of adjacent faces whose normals point towards each other, using the matching corners as defined by the face traversal starting from c_1 and c_2.

We have developed a prototype system for creating multi-segment curved handles based on the above algorithm. The users of the system can control the number of segments, weight values w_1 and w_2 using sliders. All the examples in this paper are created interactively using this prototype system. The usability of the system was also tested in a graduate level shape modeling course in which majority of the students had an architecture undergraduate background. Although none of the students had any idea about topology before, they easily learned the software and created a wide variety of high genus manifold meshes as one of their weekly class project. Figure 10 shows an example of how students use handle creation in modeling. In this example, the legs of the horse are constructed as multi segment curved handles. Another example of a shape created by a student had been shown in Figure 3.

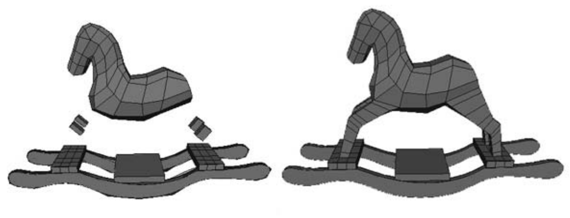


Figure 10. A manifold horse model created by a student. The legs of the horse are constructed as multi segment curved handle by connecting manifold meshes.

5 Role of Subdivision

Subdivision schemes play an important role in our approach. Our system provides various subdivision schemes including Catmull-Clark [9] and Doo-Sabin [11]. Subdivision schemes and our approach to constructing multi-segment curved handles support each other in various ways.

1. *Control mesh modeling.* Subdivision control meshes can easily be modeled by our handle construction method. The shapes in Figure 11A show two examples of high-genus control meshes that are constructed starting from a simple convex polyhedron. The shapes in Figure 11B are obtained by applying Catmull-Clark subdivision to the shapes in Figure 11A. Control shapes for knots can also be obtained with multi-segment handles. Figure 12 shows how to create a control shape starting from four identical cubes. The control shape shown in Figure 12B is obtained by connecting these four cubes with multi-segment curved handles as shown in Figure 12A. The Figures 12C and D show two views of the knot surface that is obtained by applying Catmull-Clark subdivision [9] to the control shape.

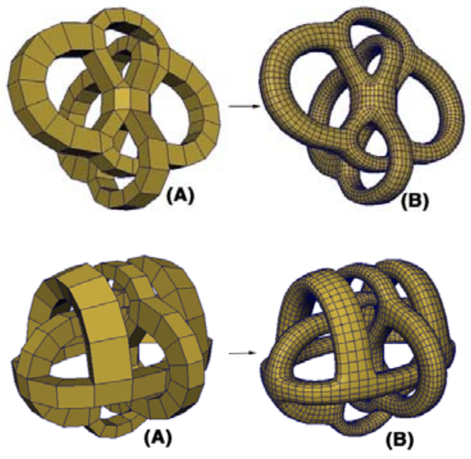


Figure 11. Two examples of manifold meshes created by our method and smoothed by Catmull-Clark subdivision [9].

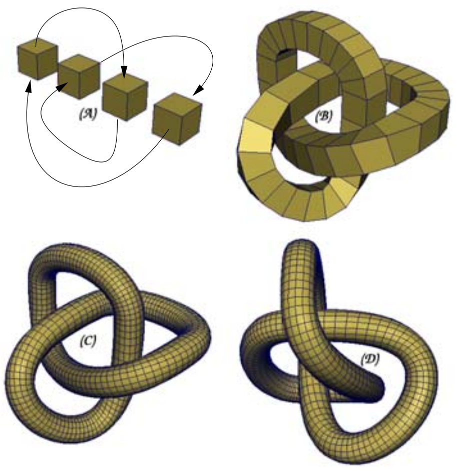
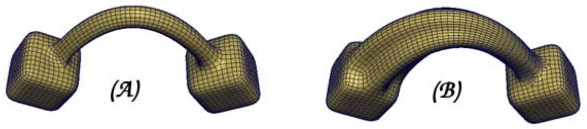


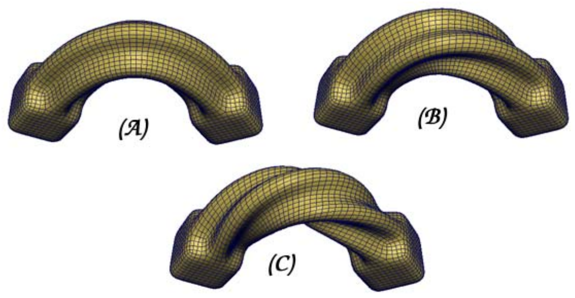
Figure 12. A knot surface created by connecting four cubes.

2. *Handle improvement.* With the usage of subdivision, we do not have to use a high number of segments to create high resolution and smooth handles. In fact, handles with a very high number of segments that are

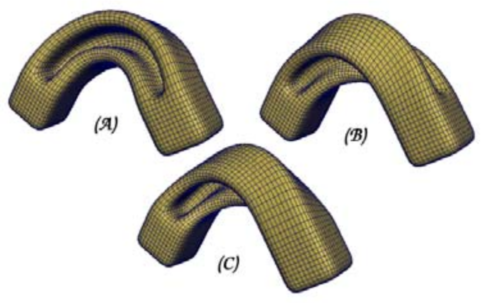
part of low-resolution meshes may not be desirable in most cases. Such handles can create an inconsistency between the resolution of the handle and the rest of the shape. Therefore, it is better to first construct a handle with a reasonable number of segments and then apply a subdivision algorithm to increase the resolution of the overall shape. For this purpose, subdivision smoothing schemes are extremely useful to improve the quality of the handles. For instance, the examples in Figures 7, 8, and 9 give an idea of how the handles are constructed consistent with the rest of the mesh, but the handle shapes are not easily perceived because of the low number of segments. Figure 13 show smoothed versions of the shapes shown in Figures 7, 8, and 9 respectively. The shapes of the handles become more visible after the subdivision as shown in Figure 13.



Smoothed shapes from Figure 7



Smoothed shapes from Figure 8



Smoothed shapes from Figure 9

Figure 13. Examples of handle improvement with subdivision. In these examples, we applied Catmull-Clark twice to the original meshes.

6 Discussion, Conclusion and Future Work

In this paper we have demonstrated an algorithm to create multi-segment, curved handles between two distinct faces of an orientable 2-manifold mesh. The algorithm allows us to create high genus, smooth surfaces. The handle creation process is robust and produces a smooth transition from one face to another without any self intersections.

Although the algorithm can only work on orientable surfaces, non-orientable looking surfaces can easily be obtained with the method. Figure 14 shows the creation of twice twisted Möbius-like surface by connecting two prisms. Figure 3 also showed one such example. Figure 15 shows another non-orientable like shapes, a Klein-bottle that is constructed by (1) duplicating one manifold surface, (2) reversing the normals of the second surface, and (3) connecting two surfaces with a curved handle.

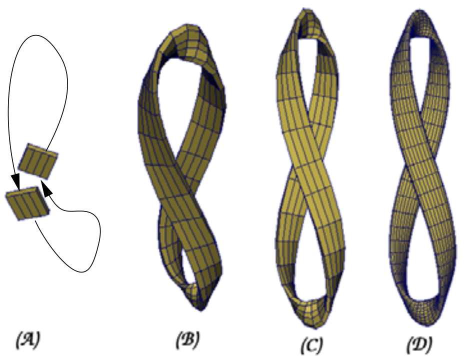


Figure 14. A Möbius-like surface with two twists created by our approach.

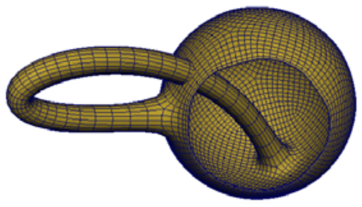


Figure 15. A Klein Bottle like surface created by our approach. We cut a portion of the surface to show the structure.

Although the morphing algorithm guarantees non-self-intersection only for star shaped faces, it is still possible to use the algorithm for the cases such as the ones shown in Figure 9. Using a 2D morphing scheme such as the one described in [17, 18, 8] it is possible to handle more general non-star shaped faces. Moreover, some of the advanced algorithms used for extruded or swept surfaces can also be used to create better sampling of segments to avoid self intersections. Although there is no need to use more complicated morphing and extrusion schemes for interactive applications, for off-line applications any elaborate morphing or extrusion scheme can easily be implemented into the current framework without a major change.

One of the biggest remaining challenges is to be able to construct handles from multiple faces to multiple faces. A specific case of the problem "one to many curved handles" is in modeling trees. Unfortunately, trees cannot be modeled with our current approach since branches are generally tangent (not perpendicular) to the trunks of the trees. This problem can be solved with a more general curves than Hermitian curves defined by face normals. Such an extension requires the development of user friendly methods that would allow direct control of the shape and properties of the curved handles.

If a smoothing subdivision scheme is applied after connecting the nearest faces of two surfaces with a handle, the resulting shapes strongly resemble smoothly blended implicit surfaces. Such implicit surfaces are essential for modeling the shapes such as chemical molecules [6, 22] and topologically changing shapes such as water particles [7]. A future application of combination of this method with subdivision schemes can be representing such smoothly blended and topologically changing shapes.

References

[1] E. Akleman and J. Chen, "Guaranteeing the 2-Manifold Property for meshes with Doubly Linked Face List", *International Journal of Shape Modeling* Volume 5, No 2, pp. 149-177, 2000.

[2] E. Akleman, J. Chen, and V. Srinivasan, "A New Paradigm for Changing Topology During Subdivision Modeling," *Pacific Graphics 2000*, October 2000, pp. 192-201.

[3] E. Akleman, J. Chen, F. Eryoldas and V. Srinivasan, "A New Corner Cutting Scheme with Tension and Handle-Face Reconstruction," *International Journal of Shape Modeling* Volume 7, No 2, pp. 111-128, 2001.

[4] E. Akleman, J. Chen, and V. Srinivasan, , "A Prototype System for Robust,Interactive and User-Friendly Modeling of Orientable 2-Manifold Meshes," *Proceedings of International on Shape Modeling and Applications 2002* (SMI'02), (2002), pp. 43-50..

[5] J. Chen and E. Akleman, "Topologically Robust Mesh Modeling: Concepts, Data Structures and Operations," Manuscript available at http://www-viz.tamu.edu/faculty/ergun/research/topology/tr02a.ps.gz

[6] J. I. Blinn, "A Generalization of Algebraic Surface Drawing", *ACM Transaction on Graphics,* vol. 1, no. 3, pp. 235-256, 1982.

[7] J. Bloomenthal, editor, *Introduction to Implicit Surfaces*, Morgan Kaufman, July 1997.

[8] E. Carmel and D. Cohen-Or, "Warp-Guided Object-Space Morphing", *The Visual Computer*, Vol. 13, 1997, pp. 465-478.

[9] E. Catmull and J. Clark, "Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes", *Computer Aided Design*, No. 10, September 1978, pp. 350-355.

[10] J. Chen, "Algorithmic Graph Embeddings", *Theoretical Computer Science*, No. 181, 1997, pp. 247-266.

[11] D. Doo and M. Sabin, "Behavior of Recursive Subdivision Surfaces Near Extraordinary Points", *Computer Aided Design*, No. 10, September 1978, pp. 356-360.

[12] M. C. Escher, *The graphic works: introduced and explained by the artist,*, (image plate 40: Band Van Möbius II), Barnes and Nobles Books, New York, 1994.

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

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

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

[16] M. Mantyla, *An Introduction to Solid Modeling*, Computer Science Press, Rockville, Ma., 1988.

[17] T. W. Sederberg and E. Greenwood, "A Physically Based Approach to 2D Shape Blending", *Computer Graphics*, No. 26, July 1992, pp. 25-34.

[18] T. W. Sederberg, P. Gao, G. Wang and H. Mu, "2D Shape Blending: An Intrinsic Solution to the Vertex Path Problem", *Computer Graphics*, No. 27, August 1993, pp. 15-18.

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

[20] W. Welch and A. Witkin, “Free-Form Shape Design Using Triangulated Surfaces”, *Computer Graphics,* **28** (August 1994) 247-256.

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

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

[23] D. Zorin and P. Schröder, editor, *Subdivision for Modeling and Animation,* ACM SIGGRAPH’2000 Course Notes no. 23, July, 2000.