EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (2009)
C. Grimm and J. J. LaViola Jr. (Editors)

Sketch-Based Subdivision Models

A. Nasri[†1] And W. Bou Karam[1] And F. Samavati[2]

[1]Computer Graphics & Animation Lab, Department Of Computer Science, American University Of Beirut,
[2]Computer Sciences Department, University Of Calgary

Abstract

Designing a control mesh (or a polyhedron) for a subdivision model is a tedious task. It involves many difficult decisions such as how to minimize the number of extraordinary vertices, how best to choose their valencies, and where to place them in the control mesh. In this paper, we present an intuitive and interactive approach for using sketch-based interface to design subdivision models. The input to the system is a set of strokes forming the profile curves of the surface. From the constructed control polygons of the sketched curves, a coarse and quad dominant control mesh is generated with few extraordinary vertices or faces. The corresponding limit surface interpolates the profile curves with the capability of local control across these curves and of the model in general. Although our approach is oriented towards quad-based systems such as Catmull-Clark, it could well be adopted in other subdivision schemes.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry—Object Modeling: Modeling packages

1. Introduction

Subdivision techniques are well established in the computer graphics industry. They provide coherent approaches for designing and modeling objects with greater flexibility alleviating the restriction of the rectangular topology of the B-spline systems. Designing the the control mesh (or polyhedron) of a model involves various decisions such as how to minimize the number of extraordinary vertices, where to place them, and how to best choose their corresponding valencies [NSY09].

Creating a 3D model using a traditional WIMP (Window, Icon, Menu, and Pointer) interface can be a very challenging and daunting task to a novice user. While such interfaces do provide the user with power, functionality and flexibility, users must invest a considerable amount of time to become proficient enough to actually produce anything meaningful. In recent years, much research has gone into the development of more accessible and natural interfaces - leading to the development of the sketch based interfaces for modeling (SBIM) complex objects [OSSJ08]. At an early stage of design, artists and designers often use pencil or pen and ink sketches to make prototypes of their ideas. Sketching a curve is the basic primitive for an artist to build preliminary drawings. The goal is then to allow users to input hand-drawn sketches which are used for the creation of 3D models along with any fine details that need to be introduced.

Since the industry standard for the creation and animation of 3D models for the last decade or so has been subdivision surfaces, it seems natural to create this kind of surfaces by a sketching process. In this paper, we aim to produce a sketch-based interface for a modeling system that allows the rapid creation of detailed models using subdivision surfaces. This would not only allow users already familiar with subdivision surfaces to adapt quickly, but also allow easy integration with already existing modeling packages in the industry (such as Maya, 3D Max, and others). The first attempt along this direction was the work of Beets et al [KJF06]. They described a sketching system for subdivision surfaces based on hybrid quad/triangle scheme. The surface generated are mainly C^1 surfaces with bounded curvature at the quad/triangle border but not the extraordinary vertices.

In this paper, we introduce a new sketch-based subdivi-

† Corresponding Author. This work was supported in part by a URB grant number #Ddf 111135 988109 from the American University of Beirut and a grant number #111135-522291 from the Lebanese National Council For Scientific Research.

Sketch-Based Interfaces and Modeling 2009, New Orleans, LA, August 1–2, 2009.
© 2009 ACM 978-1-60558-602-1/09/0008 $10.00

sion system capable of producing high quality surfaces with various features along the input profile curves. Our system generates coarser and quad dominant control meshes with a few extraordinary vertices/faces. Being subdivision models, the quality of our surface depends on the employed subdivision scheme. In the Catmull-Clark setting, it is C^2 cubic B-spline everywhere except at the extraordinary vertices. Furthermore, our models interpolate the sketched profile curves. Meeting such constraints not only plays a critical role in faithfully satisfying the intent of the artist/desinger but also allows for additional features along these curves including sharp cross curvature. These curves could also be defined by multiple strokes making the system more natural to use. To obtain their corresponding B-spline control polygons, we follow a similar approach to Pusch et al [PSNW07] using reverse subdivision but with the cubic filter described in [SBO07]. Finally, by decomposing the mesh into regular areas where n of them could be blended at an irregular region, called a *docking* station, our system maintain the local modification property and limit the extraordinary vertices/faces.

The paper is structured as follows. Section 2 gives a brief overview of related work, while Section 3 outlines the use of polygonal complexes in curve interpolation by subdivision surfaces. Section 4 formulates the problem and describes how to construct a control mesh from a set of given profile curves. Finally, Section 5 discusses preliminary results whereas conclusions and future work are described in Section 6.

2. Related Works

For the sake of limited space, we skip the details of subdivision surfaces and the Catmull-Clark scheme [CC98]. For more details, the reader is referred to [ZS00, WW02] and the references cited therein.

Several techniques have already been developed for the creation of models using sketch based interfaces, each suffering from certain limitations [OSSJ08]. For example, the SBIM Teddy system developed in [IMT99] creates triangular meshes out of 2D silhouettes and supports many useful operations, such as extrusions, cutting/tunneling, and surface augmentations. However, the quality of the meshes created in Teddy was quite poor, leading to some visual artifacts (such as bumps and dents). To overcome this, Smooth Teddy was developed in [IH03] using a fairing technique to smooth out the resulting surface. Another, more recent work developed in FiberMesh [NISA07] casts the problem as a nonlinear optimization problem that generates a fair surface fitting the input sketch. It also supports sharp features on the surface of the generated model as well as most of the sketch based operations developed in Teddy. Both systems (Smooth Teddy and Fibermesh) use global optimization to determine the geometry of the generated shapes, sometimes causing local modifications to the sketch to have unexpected effects. The meshes are also relatively dense and are therefore unsuitable for use in a subdivision system where coarser meshes are preferred. In addition, once the meshes undergo subdivision, they will "shrink", possibly causing the mesh to diverge from the artist's intent. They also do not take curves into consideration when building their meshes, but polygonal approximations of the input curves.

Other approaches use implicit surfaces to represent the generated models, such as the approaches mentioned in [AGB04] and [SWSJ07]. Approaches that depend on implicit surfaces for model generation are usually simple to understand and implement, but are usually slow and do not allow surfaces with sharp features. Another method used in [CSSJ05] and [EHBE97] uses parametric surfaces such as NURBS patches, surfaces of revolution and rotational blending surfaces which are well studied representations and easy to integrate into an application. However, the simple parameter space limits the topology of a single surface to shapes homeomorphic to a sphere. Creating more complex objects (branching object for example, or objects with multiple holes) would require crude patch intersections or careful alignment of several patches. Yet another approach taken by several authors ([TBSR04], [LBKS07], [SI07], and [Rub91]) involves shape recognition in which a sketched object is replaced by a model from a database that most closely resembles it. Although this approach is effective is some fields such as architecture or interior design, it is not general or flexible enough, since it largely depends on the richness of the model database and limits user creativity in the sketched scene.

Finally, the approach in [KJF06] creates control meshes from 2D sketches and uses the Quad/Triangle subdivision scheme developed in [SL03] to generate a smooth surface. Their limit surfaces are mainly C^1 except at the extraordinary vertices and do not interpolate the sketched curves.

3. Catmull-Clark Polygonal Complexes

In this section we give a brief overview of interpolating curves by Catmull-Clark subdivision surfaces using polygonal complexes.

Polygonal complexes were introduced by Nasri [Nas00] in the context of interpolating curves by subdivision surfaces. Such a complex is simply a control mesh which under subdivision converges to a curve. The key point is that when a complex is embodied within a control mesh, its limit curve will be automatically interpolated by the limit surface of this mesh. The topology of the mesh depends mainly on the subdivision scheme to be employed. In the Catmull-Clark setting, a polygonal complex is a set of triplets (V_b^i, V_m^i, V_t^i) in 3D space connected by a middle control polygon (V_m), see Fig. 1. It was shown [NA02] that the limit of a Catmull-Clark (CC) complex is the cubic B-spline curve defined by vertices C_i (colored red in the Figure) given by:

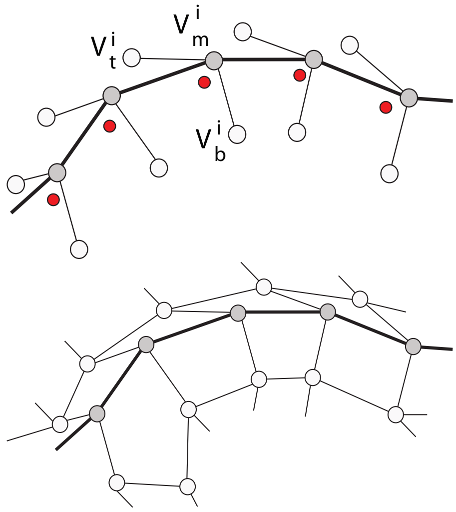

Figure 1: *A Catmull-Clark polygonal complex (top) defined by the triplets* (V_b^i, V_m^i, V_t^i) *and the same complex embodied in a control mesh (bottom). The red vertices (top) are the control vertices of the limit curve of the complex.*

$$C_i = \frac{V_b^i + 4V_m^i + V_t^i}{6} \tag{1}$$

If the middle control vertices (V_m^i) are tagged to form a control polygon whose limit curve is to be interpolated by the surface, then this could be achieved by simply replacing each vertex V_m^i by W_m^i given by:

$$W_m^i = \frac{-V_t^i + V_m^i - V_b^i}{4} \tag{2}$$

The advantage of using polygonal complex in curve interpolation is the capability of interpolating additional features such as tangent or even curvature across the interpolated curves [NSAZ*06].

4. The Control Mesh Generation

Our problem could then be formulated as follows. Given a set of strokes sketching the profile curves of the subdivision model to be designed, we need to create a control mesh for this model whose limit surface interpolates these curves. In order to explain our approach, we consider a closed profile curve. Open curves should be easily inspired.

As in every mesh construction process, our algorithm consists of two main phases: the topology phase and the geometry phase. In the former, we determine the number of ordinary and extraordinary vertices and the faces forming the control mesh, whereas in the second phase we position the vertices of the model in the 3D space. We begin by the topology phase.

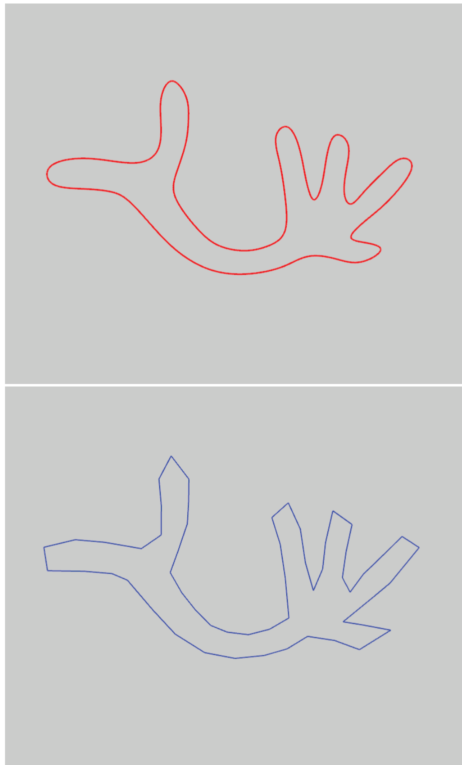

Figure 2: *A sketched profile(top) and the control polygon by Reverse Cubic Filter (bottom).*

4.1. The Topology Phase

Among the constraints that dictate the number of extraordinary vertices and their connectivity are the profile interpolation by the limit surface of the model and the local modification of the surface in particular for extrusion. To satisfy the first requirement, we avoid extraordinary vertices along the profile curve to allow for the polygonal complex construction and to control the shape of the model across the interpolated curve.

To illustrate the steps involved in our approach, we use the simple rofile curve given in Fig. 2. The following steps are then in order:

The Control Polygon Generation The stroke's points, directly created from an input device, cannot be used for constructing the mesh due to several problems. For instance, the points are noisy because of the shaky nature of handling the input devices. In addition, there will be a very large number of points because the input device sends data many times per second. To create an appropriate control mesh for Catmull-Clark subdivision, we need

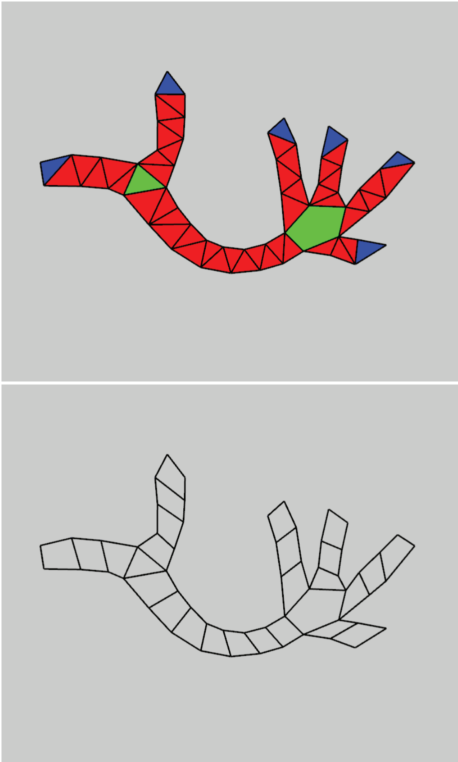

Figure 3: *The constrained Delauny triangulation of the control, polygon in Fig. 2 and the corresponding quadrangulation of the enclosed region.*

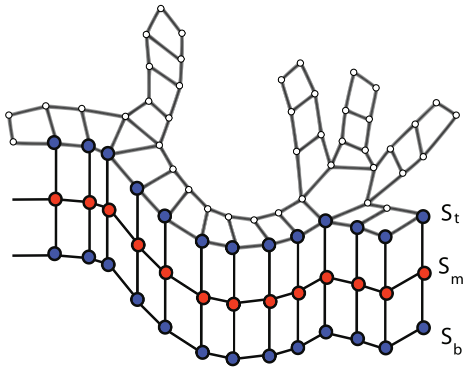

Figure 4: *Topological connection of the two layers: Partial connection of top and bottom layers with the middle one is shown. The red vertices are the control polygon of the sketched profile curve.*

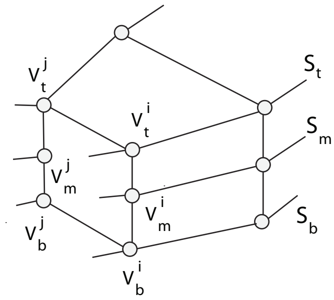

Figure 5: *Geometric positioning of the layers. Every couple of vertices (V_t^i, V_b^i) and their counterpart (V_t^j, V_b^j) are initially positioned symmetrically about the middle vertices V_m^i and V_m^j, respectively, at a half the distance between $V_m^i V_m^j$.*

to reduce and denoise the input points. Although classical approaches can be used for filtering the noise and simplifying the stroke in separate steps, we are interested in employing a technique that is consistent with the subdivision framework. Particularly, we are interested to find a coarse approximation of the stroke such that its refinement after using Catmull-Clark subdivision becomes very close to the given stroke. In fact, it is desired to reverse the act of Catmull-Clark subdivision on the stroke's points. In general, the reverse subdivision techniques [SB99, BS00] can address all of these issues. In this approach, the set of high resolution points is efficiently decomposed into a low resolution approximation and a set of details vectors. Removing details can filter the noise and the low resolution approximation can be used to reduce the points. Since Catmull-Clark subdivision is an extension of bi-cubic B-spline subdivision, we use reverse of cubic B-spline subdivision [SBO07, BS00].

Fig. 2 shows an example of a sketched profile curve and its corresponding control polygon.

The Constraint Delaunay Triangulation (CDT) Let (V_i) be the vertices of the resultant control polygon P. To capture the topology of the region enclosed by the set of vertices (V_i), we triangulate it using using a CDT algorithm [She96]. Each triangle of the resulting mesh is generally classified into one of the three categories:

1. Terminal triangle: This is a triangle with one internal edge. Such a triangle is indicated by blue color on Fig. 3.
2. Sleeve triangle: that is a triangle with two internal edges. Such a triangle is indicated by a red color on the same Figure.
3. Branching triangle: that is triangle with all edges being internal. These triangles are merged into a region with green color as suggested below.

Fig. 3 depicts the CDT of the corresponding control polygon shown in Fig. 2.

The 2D Mesh Topology Our subdivision model could be

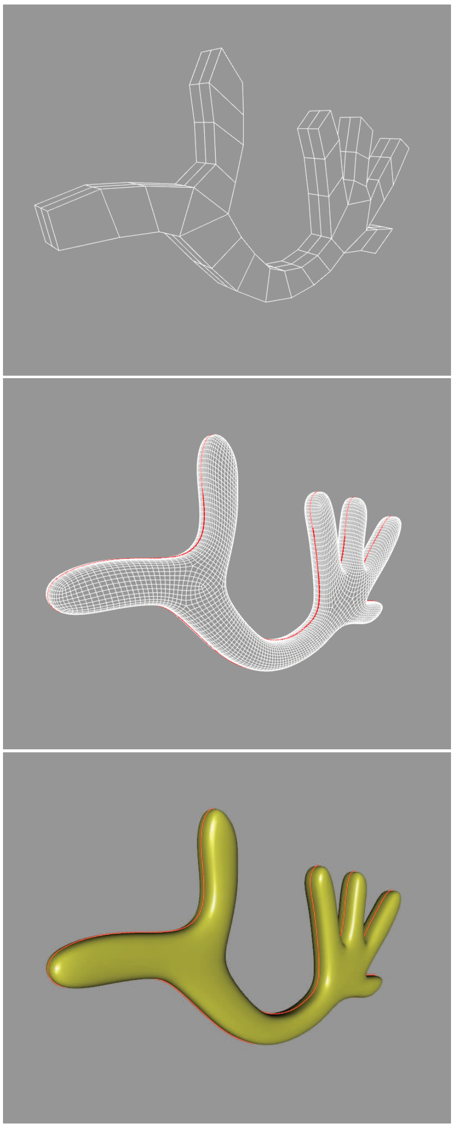

Figure 6: *The control mesh and the limit model from the curves in Fig. 2*

thought of as a set of sleeves where at most n of them are blended together by an n-sided region. Our goal is then to decompose the triangulated region into a similar set of sleeves but in 2D.

This process begins by merging all branching triangles into an n-sided face, called *an n-sided branching face*, shown in green color on Fig. 3. Accordingly, the branching faces decompose the set of triangles T into subsets (T_r), each between two consecutive branching faces, two terminal triangles, or one of each. Ideally, we want each subset T_r to have an even number of triangles, but if it does not, the side of T_r with less vertices is re-sampled by adding one extra vertex to make the number of triangles even. This helps in reducing the number of extraordinary vertices/faces on the generated mesh. Once this is accomplished, the triangles of a sleeve are used as the basis of our quads. Basically, we start from the triangle of T_r which has an adjacent edge with the branching face and collapse every two adjacent triangles into one quad. As such, a *branching* sleeve becomes a strip of quads with every two adjacent quads share an edge, except the *end-quads*. The shared edges are topologically speaking parallel to the side of the branching faces to which the sleeve is attached. Accordingly, an n-sided branching face acts as a *docking* station (see Fig. 3) to which at most n branching sleeves could be attached. It is noteworthy to mention that as extraordinary faces/vertices are almost unavoidable at branching regions, the use of docking stations limit these extraordinary elements to its vicinity and the mesh everywhere else is almost regular.

The above analysis lays down the foundation based on which more complicated topological structure could be constructed as explained in Sect. 5.

The 3D Mesh Topology Topologically, the control mesh consists of two copies of the planar mesh generated above: S_t with vertices (V_t^i) and S_b with vertices (V_b^i). These layers are placed above and below a control polygon of vertices (V_m^i) which is a duplicate of the control polygon obtained by the reverse subdivision process. In the rest of sequel we denote this control polygon by S_m and its vertices by $V_m^i = V_i$.

The mesh construction consists then of connecting the vertices of the boundaries of S_t and S_b via the control polygon S_m as shown in Fig. 5. The vertices (V_t^i), (V_m^i), and (V_b^i) form then the set of triplets of the polygonal complex needed to interpolate the profile curve defined by S_m.

4.2. The Geometry Phase

The most important part in this phase is how to position the triplets of the Catmull-Clark complex which under subdivision converges to the profile curve.

To begin with, let us consider one of those triplets (V_t^i, V_m^i, V_b^i) and assume that its opposite triplet is (V_t^j, V_m^j, V_b^j), as indicated in Fig. 5. Initially, the top and bottom vertices (V_t^i, V_b^i) of the triplet are positioned symmetrically about the middle vertex V_m^i at half the distant of $V_m^i V_m^j$. The vertices of the opposite triplets are also positioned in a similar manner. This is repeated for all triplets of the two layers giving an initial 3D polyhedron with an embodied polygonal complex (V_t^i, V_m^i, V_b^i). In general, the vertices of the middle row of a complex must be repositioned in order to interpolate the curve defined by (V_m^i), as described in Sect. 3. However, since (V_m^j) is the midpoint of $V_t^j V_b^j$, Eq. 1 is satisfied and the profile curve is interpolated automatically. With

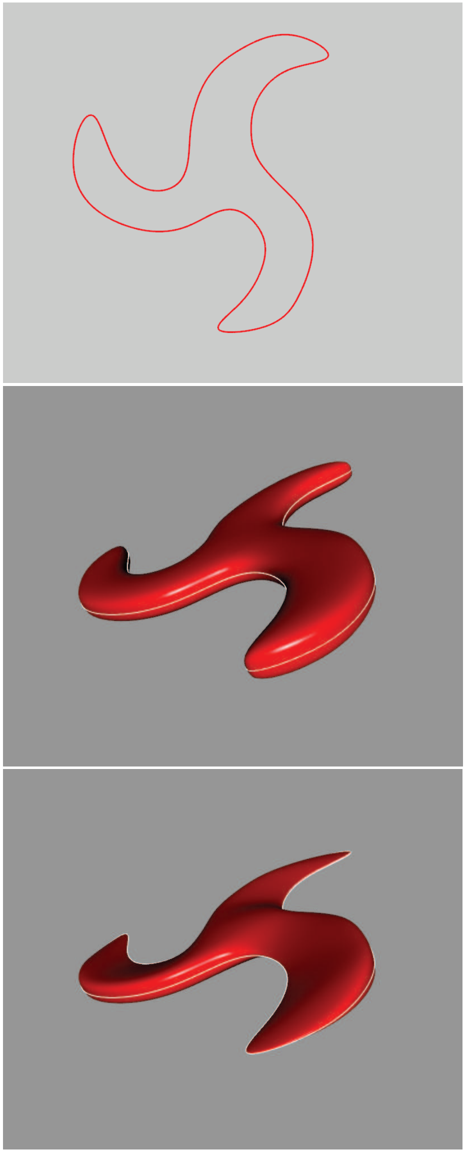

Figure 7: *A 3-blad fan without (top) and with sharp features across the profile curves.*

this said, the vertices of two layers (V_t^i, V_b^i) do not have to be necessarily equidistant to the vertices (V_m^i) of the middle control polygon, nor collinear. In such a situation, each vertex V_m^j is to be replaced by W_m^j as in Eq. 2. Furthermore, the user could manipulate the position of these vertices to locally control the shape of the model across the profile as long as the above relations are maintained.

5. Results and Discussions

The proposed approach is implemented to create control meshes for the Catmull-Clark subdivision models in Figs. 6–9. For each of these models, the sketched profile curves and the constructed control mesh are shown. In our system, we have found that four iterations of the reverse subdivision seem to yield the best results for the control polygon of the profile curves. This is the default setting used in all the curves displayed here, and the user is left with the option to smooth out the curve using further iterations, or apply non-shrinking Gaussian smoothing (see [Tau95]).

As can be seen, our system could handle many branching pieces where the extraordinary vertices or faces are mainly introduced there. The system uses the fundamental idea of this docking station to blend at most n sleeves. It is noteworthy to mention that in complicated configurations, any quad could become a branching face to which other sleeves could be attached too. In other situations, in particular for sketched curves with high curvature, a sleeve could consist of one quad only or even rounded to make a hole as shown in the Fig. 8. It is also common to have branching faces adjacent to other branching faces or even a quad with 2-valent vertex. Moreover, our algorithm could also handle open profile curves.

Being limit of subdivision, our models are smooth and of good quality. The degree of the smoothness depends mainly on the employed subdivision scheme. For instance, in Catmull-Clark setting, the surface is C^2 cubic B-spline everywhere except at the extraordinary vertices where bounded curvature is also achieved. The quality of our surface is improved by limiting the number of extraordinary regions. Our system avoids generating any extraordinary vertex along the path of a profile curve. Not only this is necessary to satisfy the interpolation constraints, but also needed to achieve the often desired C^2 continuity of the profile curves.

Moreover, our scheme interpolates the input sketched curves. Meeting such constraints plays a critical role in faithfully satisfying the intent of the artist/desinger. The curves could also be used as shape handles to locally modify the curvature across the profile curves where various features along them could be achieved using the polygonal complexes; see Fig. 7 where a 3-bladed fan with sharp edges is shown.

The docking stations used above decompose our mesh into independent sleeves which could be locally modified without affecting other sleeves. This is an important feature for adding more operations such as extrusion.

6. Conclusion and Future Work

We have presented an algorithm for building subdivision models from sketched profile curves. Not only does this facilitate the design process, being sketch-based, but also provides an efficient interface for building control meshes for subdivision models. The generated control meshes are mainly quad except at the branching regions. The generated objects could be manipulated by the profile curves with the possibility of additional features such as the manipulation of the curvature across the profile curves. As presented, the algorithm restricts the sleeves to have even number of triangles to facilitate the quad generation process. In order to alleviate this restriction, we are investigating a different approach for automatically generating these quads without the constrained Delauny triangulation process. As future work, we are implementing most of the features of a sketch-based subdivision system. This includes extrusion, cutting, union, and sharp features.

References

[AGB04] ALEXE A., GAILDRAT V., BARTHE L.: Interactive modelling from sketches using spherical implicit functions. In *AFRIGRAPH '04: Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa* (New York, NY, USA, 2004), ACM, pp. 25–34.

[BS00] BARTELS R. H., SAMAVATI F. F.: Reversing subdivision rules: Local linear conditions and observations on inner products. *Journal of Computational and Applied Mathematics 119*, 1-2 (2000), 29–67.

[CC98] CATMULL E., CLARK J.: Recursively generated b-spline surfaces on arbitrary topological meshes. 183–188.

[CSSJ05] CHERLIN J. J., SAMAVATI F., SOUSA M. C., JORGE J. A.: Sketch-based modeling with few strokes. In *SCCG '05: Proceedings of the 21st spring conference on Computer graphics* (New York, NY, USA, 2005), ACM, pp. 137–145.

[EHBE97] EGGLI L., HSU C.-Y., BRUDERLIN B. D., ELBER G.: Inferring 3d models from freehand sketches and constraints. *Computer-Aided Design 29*, 2 (February 1997), 101–112.

[IH03] IGARASHI T., HUGHES J. F.: Smooth meshes for sketch-based freeform modeling. In *I3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics* (New York, NY, USA, 2003), ACM, pp. 139–142.

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 409–416.

[KJF06] KOEN B., JOHAN C., FRANK V. R.: Optimizing mesh construction for quad/triangle schemes. In *Computer Graphics International* (2006), Nishita T., Peng Q., Seidel H.-P., (Eds.), vol. 4035 of *Lecture Notes in Computer Science*, Springer, pp. 711–718.

[LBKS07] LEE W., BURAK KARA L., STAHOVICH T. F.: An efficient graph-based recognizer for hand-drawn symbols. *Comput. Graph. 31*, 4 (2007), 554–567.

[NA02] NASRI A., ABBAS A.: Designing catmull-clark subdivision surfaces with curve interpolation constraints. *Computer & Graphics 26*, 3 (2002), 393–400.

[Nas00] NASRI A.: Recursive subdivision of polygonal complexes and its applications in CAGD. *Computer Aided Geometric Design 17* (2000), 595–619. Presented also at The 5*th* Siam Conferene On Geometric Design, Nashville, 1997.

[NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. Graph. 26*, 3 (2007), 41.

[NSAZ*06] NASRI A., SABIN M., ABU ZAKI R., NASSIRI N., SANTINA R.: Feature curves with cross curvature control on catmull-clark subdivision surfaces. In *Computer Graphics International 2006* (2006), H.P. Seidel T. N., Peng Q., (Eds.), vol. 4035 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 761–768.

[NSY09] NASRI A., SABIN M., YASSEEN Z.: Filling n–sided holes for subdivision surfaces. *Computer Graphics Forum (to appear)* (2009).

[OSSJ08] OLSEN L., SAMAVATI F., SOUSA M., JORGE J.: Sketch-based modeling: A survey. *Computers & Graphics 33* (2008).

[PSNW07] PUSCH R., SAMAVATI F., NASRI A., WYVILL B.: Improving the sketch-based interface: Forming curves from many small strokes. *The Visual Computer 23*, 9-11 (September 2007), 955–962.

[Rub91] RUBINE D.: Specifying gestures by example. *SIGGRAPH Comput. Graph. 25*, 4 (1991), 329–337.

[SB99] SAMAVATI F. F., BARTELS R. H.: Multiresolution curve and surface representation by reversing subdivision rules. *Computer Graphics Forum 18*, 2 (1999), 97–120.

[SBO07] SAMAVATI F., BARTELS R., OLSEN L.: *Local B-spline Multiresolution with Examples in Iris Synthesis and Volumetric Rendering*. 2007.

[She96] SHEWCHUK J. R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied Computational Geometry: Towards Geometric Engineering* (May 1996), Lin M. C., Manocha D., (Eds.), vol. 1148 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 203–222.

[SI07] SHIN H., IGARASHI T.: Magic canvas: interactive design of a 3-d scene prototype from freehand sketches. In *GI '07: Proceedings of Graphics Interface 2007* (New York, NY, USA, 2007), ACM, pp. 63–70.

[SL03] STAM J., LOOP C.: Quad/triangle subdivision. *Computer Graphics Forum 22*, 1 (2003), 79–85.

[SWSJ07] SCHMIDT R., WYVILL B., SOUSA M. C., JORGE J. A.: Shapeshop: sketch-based solid modeling with blobtrees. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses* (New York, NY, USA, 2007), ACM, p. 43.

[Tau95] TAUBIN G.: Curve and surface smoothing without shrinkage. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision* (Washington, DC, USA, 1995), IEEE Computer Society, p. 852.

[TBSR04] TSANG S., BALAKRISHNAN R., SINGH K., RANJAN A.: A suggestive interface for image guided 3d sketching. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2004), ACM, pp. 591–598.

[WW02] WARREN J., WEIMER H.: *Subdivision Methods For Geometric Design: A Constructive Approach*. Morgan Kaufmann, 2002.

[ZS00] ZORIN D., SCHRÖDER P.: Subdivision for modeling and animation, SIGGRAPH'00 course notes, 2000.

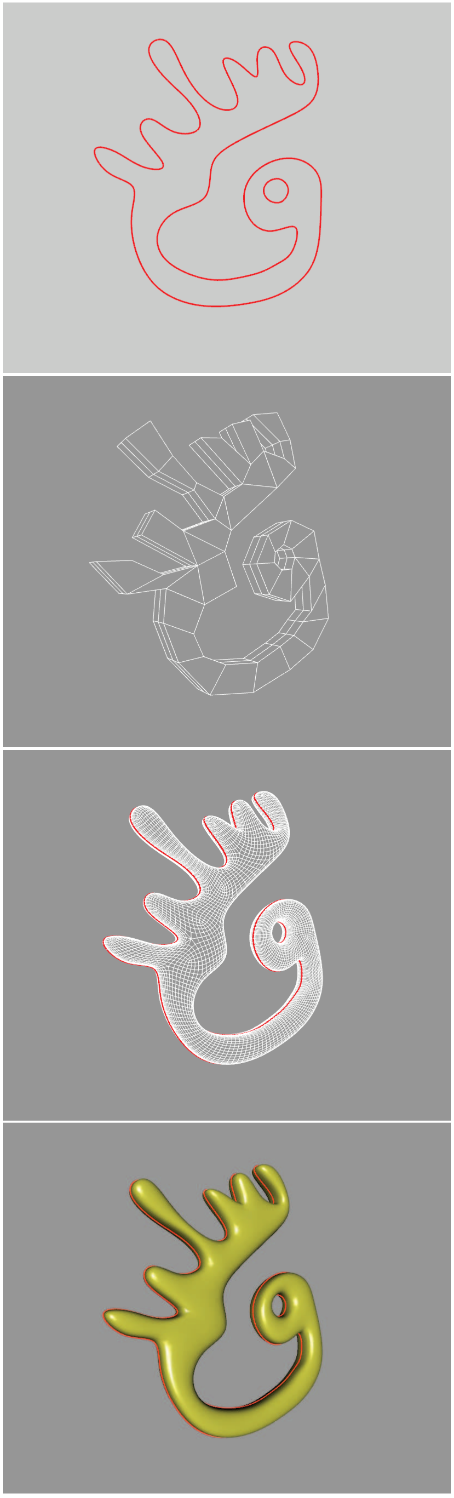

Figure 8: *A sketched profile curve with a hole and few one-quad sleeves.*

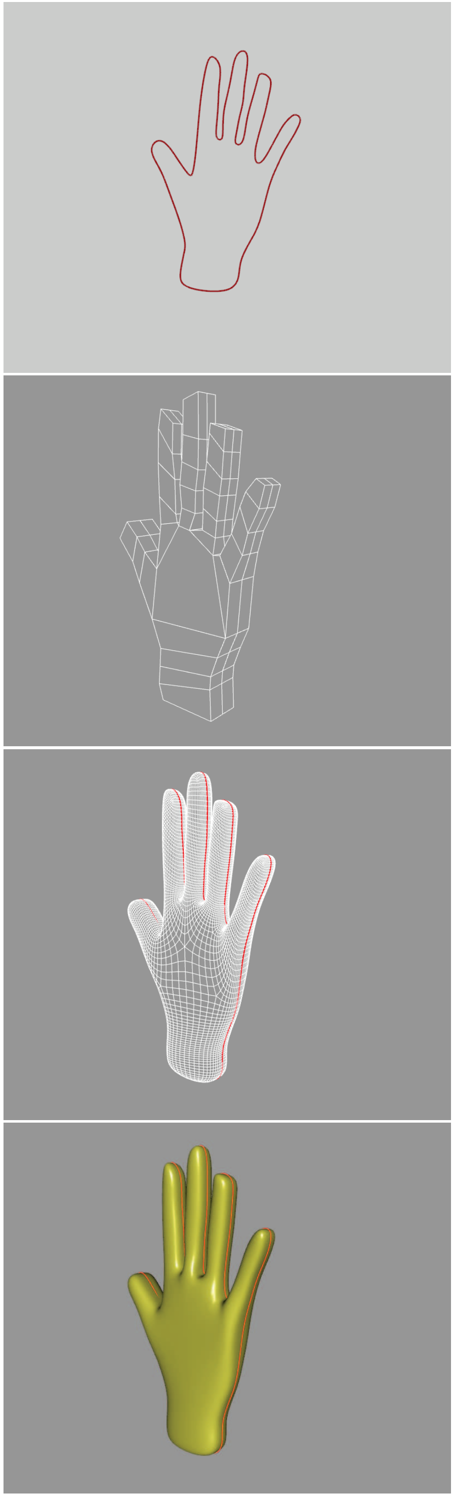

Figure 9: *A hand showing many branching fingers.*