3D Sketching with profile curves

Floriant Levet, Xavier Granier, Christophe Schlick

This is a pre-print version. The final version is available in "Lecture Notes in Computer Science" - Volume 4073/2006 (ISBN 978-3-540-36293-7).

3D Sketching with profile curves

Florian Levet, Xavier Granier, and Christophe Schlick

IPARLA project (INRIA futurs - LaBRI),
UMR 5800, Université Bordeaux 1; 351, cours de la Libération
33405 Talence, France
{levet,granier,schlick}@labri.fr

Abstract. In recent years, 3D sketching has gained popularity as an efficient alternative to conventional 3D geometric modeling for rapid prototyping, as it allows the user to intuitively generate a large range of different shapes. In this paper, we present some sketching interactions for 3D modeling, based on a set of two different bidimensional sketches (profile curve and silhouette curve). By using these two sketches and combining them with a gesture grammar, a very large variety of shapes (including shapes with topological holes) can be easily produced by our interactive modeling environment.
Key words: Sketch-based 3D Modeling

1 Introduction

Most existing geometric modeling softwares require a lot of skill from the user, to really get the shape he wants to obtain. The complexity of conventional geometric modeling has recently led to the development of some alternative modeling techniques that can be categorized as *3D sketching*. With a sketching metaphor, users can rapidly generate a 3D prototype to illustrate the 3D object they have in mind.

3D sketching is also well suited for being used on mobile devices, such as smartphones, PDAs or tablet PCs. Even if the storage and computation resources of such devices have greatly increased in recent years, the limitation to one-handed (as the other one is usually carrying the device) stylus-based interaction, makes it difficult to adapt existing modeling software that take full advantage of the mouse+keyboard interface.

With a stylus, a drawing metaphor appears quite natural for 3D modeling tasks. Different sketching approaches have already been presented as efficient and intuitive interfaces for creating and editing 3D models. They have shown that a shape can be reconstructed either from strokes and lines [26], or curves and gestures [10, 21, 11, 4, 19]. Building upon the latter curves and gestures systems, our goal is to extend the ability of these sketching approaches, while keeping efficient and intuitive interaction.

There are four main contributions in this paper. (i) The first one is a technique to account for a profile curve to infer a 3D model – Section 4. With this approach, the user is not limited to "blobby" shapes but can easily include sharp

edges. Moreover, shapes that include topological holes can also be easily generated (see Figure 1). (ii) The second contribution is a set of improvements over the original Teddy system [10] – Section 3. (iii) The third one is a method for local and global edition of the shape by a local or global change of the profile – Section 5. (iv) Finally, we present several gesture-based interactions to create a complex shape from the profile and silhouette sketches – Section 5.

2 Previous work

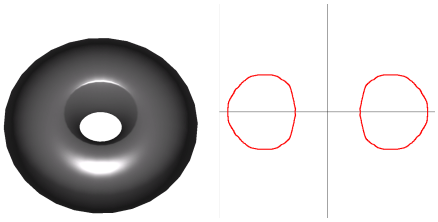
Many techniques are used by artists in order to suggest the object's shape such as characteristic lines (or contour lines) or shaded areas. Since drawing is a familiar task for a lot of people, "sketching" has been introduced as a natural alternative for 3D modeling tasks. Existing sketching approaches can be divided into three categories: the line-and-stroke approach, the painting-or-shading approach, and the curve-and-gesture approach.

The principle of the line-and-stroke approach is to infer a 3D volume from characteristic lines drawn by a user [5, 23]. Such a solution can even be interactive [15, 18]. When ambiguities occur, they can be removed by a user selection of the correct model into a list of possible reconstructions [14, 6]. Most of these approaches are limited to polyhedral models, pre-defined shapes or parametrized objects [25]. With recent methods [3, 21], more complex 3D lines can be sketched, but the final model is still limited to a wireframe one. Moreover, stroke-based approaches reduce the limitations on possible 3D lines and curves [20, 2], but they are mostly limited to illustration since they cannot really reconstruct a full 3D object.

The painting-or-shading approach allows the generation of highly detailed models. Extending the work of Williams [24], Overveld introduced a geometric modeling technique based on the painting of a surface gradient [22]. Shape editing techniques based on shading information was introduced by Rushmeier et al. [16]. More recently, 3D height-field generation by 2D shading has been presented by Kerautret et al. [12].

Currently, the most powerful techniques have been based on the curve-and-gesture approach. These techniques allow the user to create a large variety of free-form shapes [26, 10, 13] by using a gesture grammar which converts some drawn curves into a corresponding modeling operation: extrusion, deformation, cutting, etc.

Either variational surfaces [4, 11, 27], implicit surfaces [17] or convolution surfaces [1] have been used by curve-and-gesture approaches. This has the advantage to generate smooth surfaces, but also emphasizes a "blobby" aspect for the resulting shapes. To reduce this blobby aspect, Tai et al. [19] have proposed to use a profile curve, defined in polar coordinates. Profile curves have been used previously in a sketching environment limited to generalized cylinders [7]. One nice property of using implicit surfaces for the inferred geometry is that the resulting shapes can be easily merged by using classical CSG operators. On the other hand, the main drawback is that some expensive tessellation step has to



(a) (b)

Fig. 1. A tore created with two sketches (silhouette curve and profile curve).

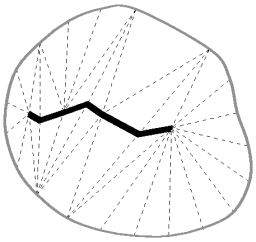


Fig. 2. Silhouette curve (grey), reconstructed skeleton (bold black) and corresponding triangulation (dot black).

be employed to convert the surface into a set of triangles that can be sent to the graphics hardware, and complex shapes can require a lot of CSG operations.

Based on these observations, we have designed our sketching environment on the following criteria. First, our system will directly infer a mesh from a set of curve-and-gesture elements. In addition to speed, using a mesh offers the possibility of adapting the sampling according to the local curvature of the surface (i.e., improve the precision where needed). Second, our system will extend the grammar defined in Teddy [10] by creating a profile curve for the geometric model that can be edited either globally or locally.

3 General approach

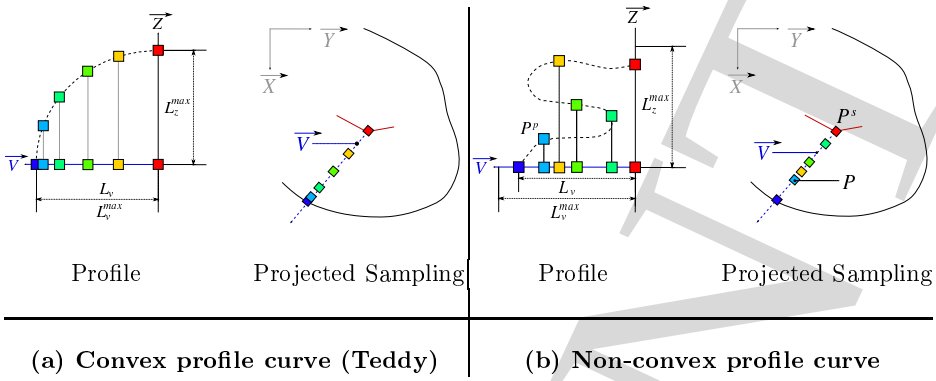
As said above, our approach is basically an extension of the Teddy system, and thus uses a similar four-step process as the one proposed by Igarashi et al [10]:

1. A silhouette curve is sketched by the user and then sampled.
2. A constrained Delaunay triangulation (CDT) is computed from the silhouette samples, and used to extract a skeleton (see Figure 2).
3. For each skeleton point, an elevation distance is computed, as the average distance to connected silhouette samples.
4. Each internal edge of the triangulation is sampled, and the corresponding points are translated according to the elevation distance, to create the final mesh vertices.

Compared to the original Teddy implementation, our system proposes two improvements of this four-step process. First, the elevation step is slightly modified, in order to account for non-convex profile curves. Second, we propose a better estimation of the normal vector at the resulting vertices, in order to obtain a smoother appearance of the final 3D object.

3.1 Elevation of internal points

With Teddy, mesh vertices are created by sampling the internal edges of the triangulation on the silhouette plane $(\overrightarrow{X}, \overrightarrow{Y})$ and elevating them according to



(a) Convex profile curve (Teddy) **(b) Non-convex profile curve**

Fig. 3. Difference between a convex and a no-convex profile curve.

a circular profile curve (see Figure 3(a)). Unfortunately, this can only work for convex profile curves (i.e., profile curves that can be expressed as a height-field).

When using non-convex profile curves, we propose to compute a set of sample points of the profile curve $P^p = (P_v^p, P_z^p)$, and use these samples along each internal edge to obtain the position of the corresponding mesh vertices $P = (P_x, P_y, P_z)$. The new elevation of a final 3D vertex is given by:

$$P_z = (P^s / L_z^{max})\, P_z^p,$$

where L_z^{max} is the maximum height of the profile curve (see Figure 3(b)). This approach guarantees that the value of the highest possible elevation is still equal to the original elevation P^s of the corresponding skeleton point. Note that, depending on the profile curve, the elevation distance may be null for some skeleton points, which mean that topological holes may be created along the skeleton, on the resulting 3D model.

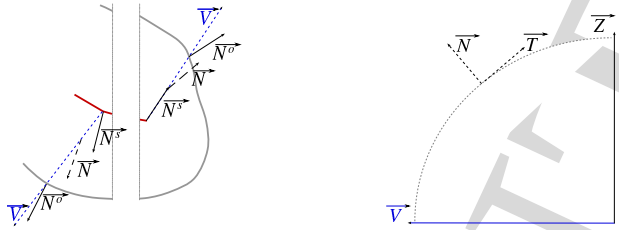
We also need to ensure that the first sample for the profile curve does correspond to a silhouette point. This can be obtained by computing the two other coordinates P_x and P_y as:

$$P_x = P_x^s + V_x \times (P_v^p / L_v) \quad \text{and} \quad P_y = P_y^s + V_y \times (P_v^p / L_v)\,.$$

Note that with convex profile curves as in Teddy, we always have the following condition: $L_v = L_v^{max}$ (see Figure 3(a)).

3.2 Improving the appearance of the model

One limitation of the original Teddy approach is the lack of smoothness of the resulting shape. To get a pleasant appearance, a smoothing step is generally required [9], which slows down the sketching process. In our approach, we propose to directly enhance the appearance during the creation of the vertices, by using an ad-hock process to estimate normal vectors. This process is based on the following observations: (i) a visual continuity is required on the silhouette points



(a) On silhouette plane (b) On profile curve

Fig. 4. Geometric configuration for normal generation

(i.e., the projection of the normal on the silhouette plane has to be collinear with the normal vector to the silhouette); (ii) at non-extremal skeleton points, the projection of the normal on the silhouette plane has to be collinear with the normal vector to the skeleton. (iii) the normal to the surface has to be orthogonal to the profile curve.

Our normal vector computation is totally local, which means that it can be done simultaneously to the vertex generation. In order to guarantee that the two first conditions are fulfilled, a linear interpolation is done between the normal to the silhouette $\overrightarrow{N^o}$ and the normal to the skeleton $\overrightarrow{N^s}$, on the silhouette plane and along each internal edge (see Figure 4 for the notations):

$$N_x = (1-\rho)N_x^s + \rho N_x^o \quad \text{and} \quad N_y = (1-\rho)N_y^s + \rho N_y^o.$$

Note that, for the extremal points of the skeleton, we set the normal as the direction of the internal edge: $\overrightarrow{N^s} = \overrightarrow{V}$. As the normal has to be orthogonal to the profile, we have the condition $\overrightarrow{N} \cdot \overrightarrow{T} = 0$, where $\overrightarrow{T}$ is the tangent to the profile (see Figure 4). This leads to

$$N_z = (-T_x/T_z)\,(N_x V_x + N_y V_y).$$

The last step is simply a unit-length normalization of the resulting vector (N_x, N_y, N_z).

4 Profile generation

4.1 Designing a profile curve

As said above, in order to create a geometric model with our sketching environment, the user has to design a second sketch, the profile curve, in addition to the usual silhouette curve. As many shapes include some symmetries, the user may wish to draw only a part of it. The system permits to draw either a quarter of the curve, completed with a double symmetry (see Figure 5(a)) or a half of the curve, completed with a single symmetry (see Figure 5(c)).

As can be seen in Figure 5(a) and Figure 5(c), the curves sketched by users may be unclosed. In order to generated solid (i.e., closed) objects, the profile

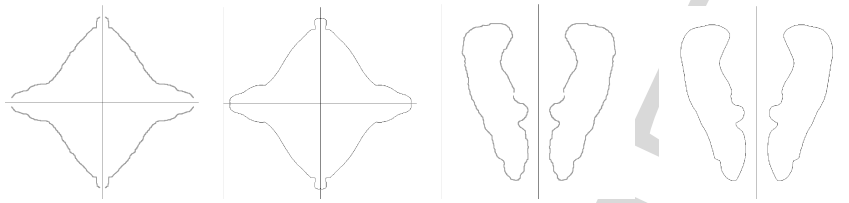
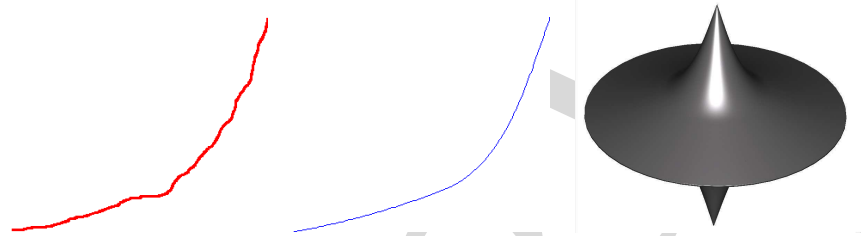


Fig. 5. (a)-(c) show two profile curves sketched by users that have been completed by symmetry. (b)-(d) show the corresponding automatically closed and smoothed curves.



(a) Input sketch of the user. (b) Smoothed profile curve. (c) Resulting model.

Fig. 6. From the input profile curve to the resulting 3D model.

curve is automatically closed after sketching, either by projecting its starting and ending points on the axes (see Figure 5(b)), or by linking its first and last points (see Figure 5(d)), according to some distance criterion.

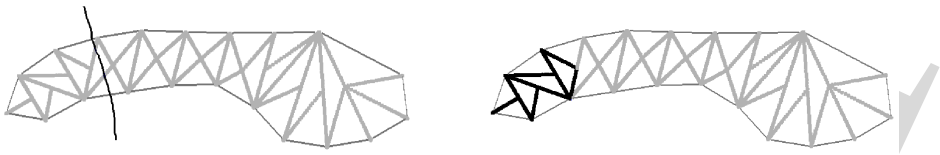
4.2 Smoothing the profile curve

As a hand-drawn sketch can be very noisy, it is usually more pleasant to smooth out high frequency features before inferring the 3D shape. One easy solution is to use some approximation spline (in our implementation, we use the NURBS++ library[1]) but any low-pass filtering technique may be employed.

Figure 6 shows the relevance of our idea. The profile curve sketched by the user is given in Figure 6(a). It is very noisy with a lot of small discontinuities. The reconstructed curve is shown in Figure 6(b). We can see that the discontinuities have disappeared and that the curve is now very smooth. The resulting 3D model on Figure 6(c) does not present the rough-and-dirty appearance often associated with sketch-based tools.

Moreover, since we are using parametric curves, we can easily control the density of their sampling, according to the local curvature (refer to Section 3 for the transformation from the sampling to the mesh vertices).

[1] http://libnurbs.sourceforge.net



(a) User has drawn a sketch of selection. (b) Selected points are in black.

Fig. 7. Local selection of a model.

5 Interaction

The design of the profile curve adds a new step in the interaction loop between users and the sketching application. Now they have to draw two sketches in order to create a model contrary to most of the sketching applications that reconstruct a model only with the silhouette sketch. So users have one more gesture to do in order to generate a model. But this gesture allows the creation of more details on the surface, and in only two gestures, it is now possible to create more complex shapes.

Since users have designed the silhouette and profile curves and the model shape was inferred, users can globally edit this shape by modifying the profile curve. When a user sketches a new profile curve, the model shape is deformed on-the-fly in order to match this new profile (see Figure 8). For a more intuitive interaction, a shape can be directly infered from the silhouette by using a default profile, and then, this profile can be globally edited.

Moreover, we have developed a local edition. This increases the range of possible shapes by enabling the combination of different profile curves along the resulting surface. In order to apply a change only locally, users have to first select a region of the surface. The system determines which internal edges have been selected (see Figure 7(b)) from a user-drawn sketch on the screen (see Figure 7(a)). Then, users sketch a new profile curve and the 3D positions of the points on the selected internal edges (i.e., a part of the object shape) are changed in order to match the new profile.

6 Results and discussion

The prototype system was written in the C++ language using the OpenGL library to render models and the Qt library to design the interface. All example models shown in this paper were built with this prototype system running on a standard desktop PC.

Our framework allows users to create a wide range of different models by only drawing two sketches. By using a default circular profile curve, any "Teddy-like" shape can be obtained (see Figure 8(a) column). By offering the possibility to

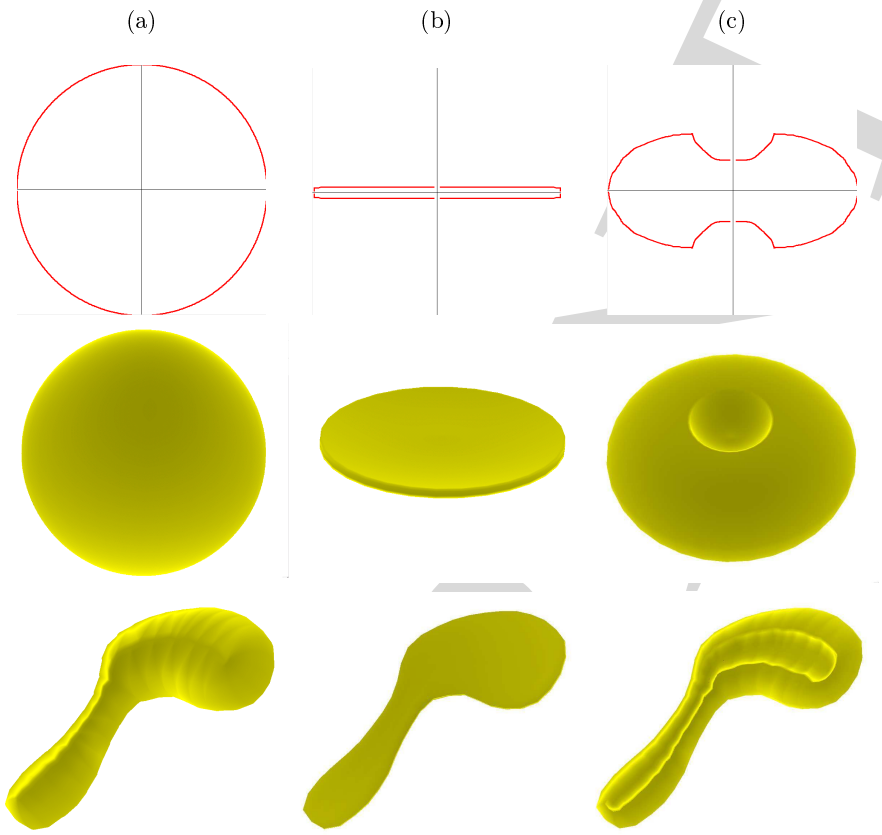


Fig. 8. Global edition of two models. (a) shows the creation of two models (one with a circular silhouette curve and the other with a more complex one) with the default circular profile curve. (b)-(c) show the effects of the global edition of the profile curve.

change the profile curve, we greatly increase the variety of shapes that can be generated (see, for instance, the flowerpot on Figure 9).

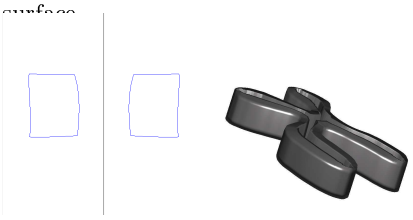
Figure 8 illustrates the principle of global edition, as discussed in Section 5. First we start from a "blobby" object by drawing a silhouette curve combined with a circular profile curve (Figure 8(a)-up). The profile curve can then be interactively modified (Figure 8(b)-up or 8(c)-up) until the user is satisfied with the resulting shape.

The modification of the profile curve may also be applied only on a part of the shape, as presented in Figure 14. The initial model was a sphere. After selecting a part of the model (in our case, the right half of the sphere), a new profile curve is designed (see Figure 14(a)) and applied on the selected part. The resulting shape is presented in Figure 14(b) and Figure 14(c).

More complex shapes can be obtained by using more complex profile curves as in Figure 13. The hammer of Figure 13(a) would require a lot of surface

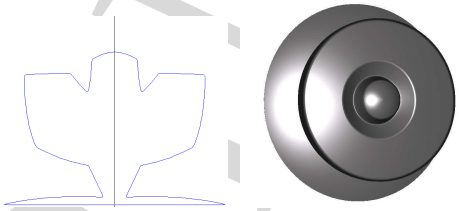
(a) (b) (c) (d)

Fig. 9. Creation of a flowerpot. This model was designed with the profile curve shown in (a) combined with a circular silhouette curve. (b) and (c) show the pot from different points of view while (d) presents a cut-view. Note that the pot is actually a genus 1 surface.



(a) (b)

Fig. 10. A model with a more complex silhouette curve.



(a) (b)

Fig. 11. Creation of a door handle with only two curves.

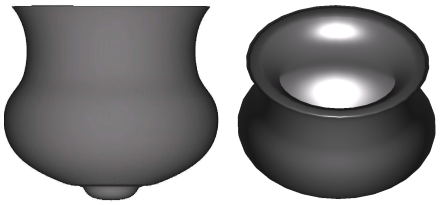
modifications in state-of-the-art sketching systems. But, with our system, we created it with only five sketches. We first sketched the hammer silhouette curve. We then did a first local selection of its head and sketched a square profile curve. Finally we selected the points of its pommel and sketched a profile curve with a crease (quite similar to the profile curve of Figure 8-upright).

Limitations

We still have some problems in the sketched shape, mostly due to the skeleton and internal points creations (these points follow the internal edges). As can be seen in Figure 15, even with simple silhouette curves, the skeletons created have a lot of small discontinuities, leading to annoying height differences for their points (as can be seen in Figure 10 and Figure 8-bottom).

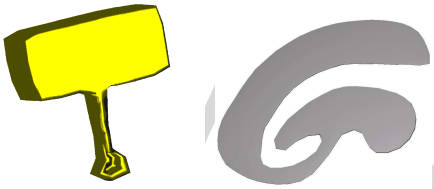
Since the creation of the mesh vertices follows the internal edges, the sampling of the model and the resulting triangulation can be under-sampled in certain regions of the model. Because more than one internal edge can end on one outline point, the resulting triangulation is far from being equilateral (with lots of thin triangles) and thus consistent evaluation of normal vectors may be difficult.

Using the approach described in [8] may be a solution to generate better skeletons without the need to use a CDT. It would require developing a new algorithm to compute internal points.



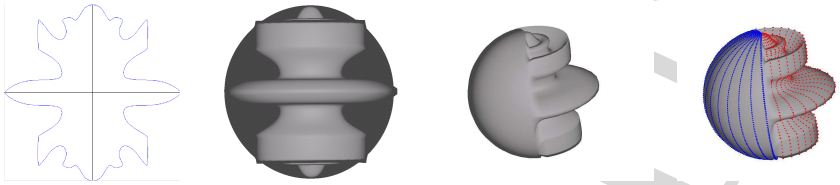
(a) (b)

Fig. 12. A vase created with two sketches.



(a) (b)

Fig. 13. More complex models created with local (a) and global (b) edition. The resulting shape in (a) is a hammer and in (b) is a letter G in 3D.



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

Fig. 14. Illustration of the local edition. The profile curve applied on the selected point is given in (a). (b-c) shows two views of the resulting model. In (d), mesh vertices are shown.

7 Conclusion and future work

In this paper, we have presented new sketching interactions for the creation of complex 3D shapes starting from a reduced number of sketched curves. A general non-convex profile curve can be combined with an arbitrary silhouette curve to generate a large variety of resulting 3D shapes.

We have also presented some ideas to improve the mesh generation from the silhouette and profile curves, by locally estimating the normal vector for each mesh vertex. Finally, after the initial mesh generation, some interactive editing, either local or global, may be used to finely adjust the final shape of the object.

Since our approach is based on the initial Teddy system, all the other gesture and curve interactions would be still usable. We haven't currently implemented all of them in our prototype software. Furthermore, a local interpolation of profile

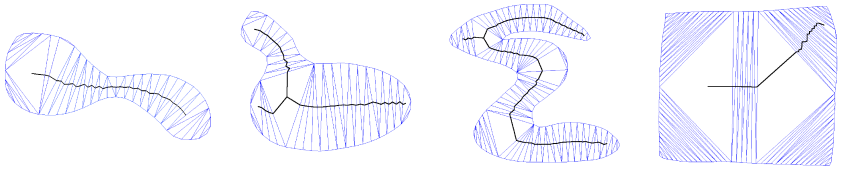


Fig. 15. CDT of different silhouette curves. We can see the skeletons discontinuities.

curves between selected zones would provide a smoother transition. This can be integrated by using curve-morphing.

The main remaining problem, as described in the Section 6, is the generation of a correct skeleton and the accurate sampling of the silhouette. These steps may lead to visual artifacts, even with our improved normal computation. We believe that a more robust algorithm for the skeleton [8], with an adaptive sampling of the silhouette may provide a solution to the problem.

References

1. A. Alexe, L. Barthe, M.P. Cani, and V. Gaildrat. Shape modelling by sketching using convolution surfaces. In *Pacific Graphics (Short Papers)*, October 2005.
2. D. Bourguignon, M.-P. Cani, and G. Drettakis. Drawing for Illustration and Annotation in 3D. *Computer Graphics Forum (Proc. Eurographics 2001)*, 20(3):114–122, September 2001.
3. Jonathan M. Cohen, Lee Markosian, Robert C. Zeleznik, John F. Hughes, and Ronen Barzel. An interface for sketching 3d curves. In *SI3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 17–21, New York, NY, USA, 1999. ACM Press.
4. A. Cuno, C. Esperança, P. Roma Cavalcanti, and R. Farias. 3D Free-form Modeling with Variational Surfaces. In *WSCG (Journal Papers) 2005*, pages 9–16, January 2005.
5. L. Eggli, C. y. Hsu, B. D. Brüderlin, and G. Elber. Inferring 3D models from freehand sketches and constraints. *Computer-Aided Design (JCAD)*, 29(2):101–112, February 1997.
6. M. J. Fonseca, A. Ferreira, and J. A. Jorge. Towards 3D Modeling using Sketches and Retrieval. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, August 2004.
7. Cindy Grimm and John Hughes. Implicit generalized cylinders using profile curves. In *Implicit Surfaces*, pages 33–41, June 1998. Creating sweep surfaces using sketching.
8. L. He. *A Comparison of Deformable Contour Methods and Model Based Approach Using Skeleton for Shape Recovery From Images*. PhD thesis, University of Cincinnati, 2003.
9. T. Igarashi and J. F. Hughes. Smooth meshes for sketch-based freeform modeling. In *SI3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 139–142, New York, NY, USA, 2003. ACM Press.
10. T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
11. O. Karpenko, J. Hughes, and R. Raskar. Free-form sketching with variational implicit surfaces. In *EUROGRAPHICS '02*, 2002.
12. Bertrand Kerautret, Xavier Granier, and Achille Braquelaire. Intuitive shape modeling by shading design. In *International Symposium on Smart Graphics*, volume 3638 of *Lecture Notes in Computer Science*, pages 163–174. Springer-Verlag GmbH, aug 2005.
13. S. Ohwada, F. Nielsen, K. Nakazawa, and T. Igarashi. A Sketching Interface for Modeling the Internal Structures of 3D Shapes. In *Proc. Smart Graphics*, pages 49–57. Springer-Verlag, July 2003.

14. J. P. Pereira, V. A. Branco, J. A. Jorge, N. F. Silva, T. D. Cardoso, and F. Nunes Ferreira. Cascading Recognizers for Ambiguous Calligraphic Interaction. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, August 2004.
15. D. Pugh. Designing solid objects using interactive sketch interpretation. In *Proc. SI3D '92*, pages 117–126. ACM Press, 1992.
16. H. Rushmeir, J. Gomes, L. Balmelli, F. Bernardi, and G. Taubin. Image-Based Object Editing. In *Proc. 3DIM '03*, pages 20–28, October 2003.
17. R. Schmidt, B. Wyvill, M.C. Sousa, and J.A. Jorge. Shapeshop: Sketch-based solid modeling with blobtrees. In *2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 53–62, 2005.
18. A. Shesh and B. Chen. SMARTPAPER–An Interactive and User-friendly Sketching System. *Computer Graphics Forum (Proc. Eurographics 2004)*, 24(3), September 2004.
19. C.-L. Tai, H. Zhang, and C.-K. Fong. Prototype modeling from sketched silhouettes based on convolution surfaces. *Computer Graphics Forum (Eurographics '04)*, 2004.
20. O. Tolba, J. Dorsey, and L. McMillan. Sketching with projective 2D strokes. In *Proc. UIST '99*, pages 149–157. ACM Press, 1999.
21. S. Tsang, R. Balakrishnan, K. Singh, and A. Ranjan. A suggestive interface for image guided 3d sketching. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 591–598, New York, NY, USA, 2004. ACM Press.
22. C. W. A. M. van Overveld. Painting gradients: free-form surface design using shading patterns. In *Proc. Graphics interface '96*, pages 151–158. Canadian Information Processing Society, 1996.
23. P. A. C. Varley, Y. Takahashi, J.Mitani, and H.Suzuki. A Two-Stage Approach for Interpreting Line Drawings of Curved Objects. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, August 2004.
24. L. Williams. 3D paint. In *Proc. SI3D '90*, pages 225–233. ACM Press, 1990.
25. Chen Yang, Dana Sharon, and Michiel van de Panne. Sketch-based modeling of parameterized objects. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 63–72, 2005.
26. R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. SKETCH: an interface for sketching 3D scenes. In *Proc. SIGGRAPH '96*, pages 163–170. ACM Press, July 1996.
27. R. Zenka and P. Slavik. New dimension for sketches. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, pages 157–163, New York, NY, USA, 2003. ACM Press.