A Sketching Interface for Modeling the Internal Structures of 3D Shapes

Shigeru Owada[1], Frank Nielsen[2], Kazuo Nakazawa[3], and Takeo Igarashi[1]

[1] The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku ,Tokyo 113-8654, JAPAN,
{ohwada|takeo}@is.s.u-tokyo.ac.jp
[2] Sony Computer Science Laboratories,Inc.,
Takanawa Muse Bldg., 3-14-13, Higashigotanda,
Shinagawa-ku, Tokyo 141-0022, JAPAN,
nielsen@csl.sony.co.jp
[3] National Cardiovascular Center,
5-7-1 Fujishiro-dai, Suita, Osaka 565-8565, JAPAN,
nakazawa@ri.ncvc.go.jp

Abstract. This paper presents a sketch-based modeling system for creating objects that have internal structures. The user input consists of hand-drawn sketches and the system automatically generates a volumetric model. The volumetric representation solves any self-intersection problems and enables the creation of models with a variety of topological structures, such as a torus or a hollow sphere. To specify internal structures, our system allows the user to cut the model temporarily and apply modeling operations to the exposed face. In addition, the user can draw multiple contours in the Create or Sweep stages. Our system also allows automatic rotation of the model so that the user does not need to perform frequent manual rotations. Our system is much simpler to implement than a surface-oriented system because no complicated mesh editing code is required. We observed that novice users could quickly create a variety of objects using our system.

1 Introduction

Geometric modeling has been a major research area in computer graphics. While there has been much progress in rendering 3D models, creating 3D objects is still a challenging task. Recently, attention has focused on sketch-based modeling systems with which the user can quickly create 3D models using simple freehand strokes rather than by specifying precise parameters for geometric objects, such as spline curves, NURBS patches, and so forth [15, 6]. However, these systems are primarily designed for specifying the external appearance of 3D shapes, and it is still difficult to design freeform models with internal structures, such as internal organs. Specifically, the existing sketch-based freeform modeling system [6] can handle 3D models only with spherical topology. This paper introduces a modeling system that can design 3D models with complex internal structures,

while maintaining the ease of use of existing sketch-based freeform modelers. We used a volumetric data structure to handle the dynamically changing topology efficiently. The volumetric model is converted to a polygonal surface and is displayed using a non-photorealistic rendering technique to facilitate creative exploration. Unlike previous systems, our system allows the user to draw nested contours to design models with internal structures. In addition, the user can cut the model temporarily and apply modeling operations to the exposed face to design internal structures. The underlying volumetric representation simplifies the implementation of such functions. Moreover, our system actively assists the user by automatically rotating the model when necessary.

The heart of our technique is automatic "guessing" of 3D geometry from 2D gestural input, and it is done by making certain assumptions about the target geometry. To be specific, the system assumes that the target geometry has a rotund, smooth (low curvature) surface [6] other than the places where the user explicitly defined the geometry by the input strokes. In other words, the user specifies the information about important features (silhouette, intersection, and sweep path) and the system supplies missing information based on the above assumption.

Our system is designed to facilitate the communication of complicated geometric information, such as surgical plans. Like other sketch-based modeling systems, however, our system is not suitable for creating the final output of any serious production, because of its lack of accuracy.

2 Previous Work

Three-dimensional shape modeling systems that use a volumetric data structure directly are relatively new [14, 4] as compared with other popular modeling primitives, such as polygons, NURBS, and subdivision surfaces. Recently, a scripting language [2], octree [11], subdivision volume [10], and level set [1] have been used as volumetric modeling methodologies. Some systems use 3D haptic input devices [4, 3, 5, 10].

Sketch-based modeling using standard mouse operations became popular in the past decade. Instead of creating precise, large-scale objects, a sketching interface provides an easy way to create a rough model to convey the user's idea quickly. One of the earliest sketching systems was Viking [12], which was designed in the context of prototypic CAD models. Later works include SKETCH [15] and Teddy [6]. The SKETCH system is intended to sketch a scene consisting of simple primitives, such as boxes and cones, while the Teddy system is designed to create rotund objects with spherical topology. Although improvements to the original Teddy system have recently been proposed [7], extending the topological variety of creatable models is still an unsolved problem.

Although the user interface of our system is based on the Teddy system, our system is free from topological limitations, provides multiple interfaces for specifying internal structures, and actively assists the user by automatically rotating a model when necessary.

3 User Interface

The entire editing operation is performed in a single window. Modeling operations are specified by freeform strokes drawn on the screen and by pressing buttons on a menu bar. The freeform strokes provide necessary geometric information and the buttons apply specific modeling operations using the strokes as input. The drawing of strokes is assigned to the left mouse button and rotating the model is assigned to the right mouse button. The current implementation uses four buttons, as shown in Fig. 1. The leftmost button is used to initialize the current scene; the second one is to create items; the third is for the extrusion/sweep function; and the last is for undo.

Fig. 1. Buttons in our system

3.1 Create

Objects are created by drawing one or more contours on the canvas and pressing the "Create" button. This operation inflates the intermediate region between the strokes leaving holes (Fig. 2).

Fig. 2. Nested contours are allowed in the Create operation.

3.2 Extrusion

Extrusion is an operation that generates a protuberance or a dent on a model. The user draws a single closed stroke on the object's surface specifying the contour (Fig. 3 (b)) and presses the "Extrusion/sweep" button. After rotating the model (Fig. 3 (c)), the user draws a second stroke specifying the silhouette of the extruded area (Fig. 3 (d, f)). The user should place each end of the silhouette

stroke close to each end of the projected surface contour (otherwise the second stroke is interpreted as a sweep path; see Section 3.4.) A protuberance is created if the second stroke is drawn on the outside of the object (Fig. 3 (d,e)). The user can also create a hole by drawing a stroke into the object (Fig. 3 (f,g)). Volumetric representation automatically prevents self-intersection problems, where specialized care must be taken when using a polygonal representation. A hidden silhouette is rendered as broken lines.

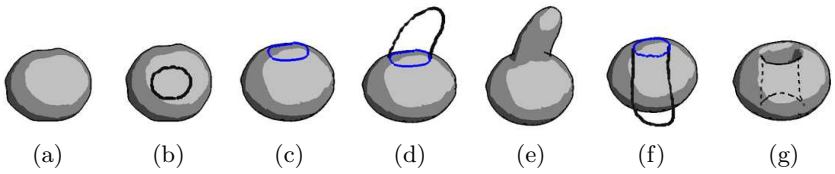

Fig. 3. Examples of Extrusion

3.3 Loop Extrusion

In addition, it is also possible to create a hollow object using extrusion. To do this, the user first cuts the model to expose the internal region (Fig. 4 (a-c)), then draws a contour on the exposed plane (Fig. 4 (d)), and finally draws a circular stroke that entirely surrounds the contour (Fig. 4 (e)). We call this operation "Loop Extrusion". The cutting operation that we use differs from the standard Cut operation in the Teddy system [6] in that the removed region is just deactivated temporarily. The system distinguishes these two operations by checking whether there is a corner at the end of a stroke. The system performs a standard cutting operation when there is no corner, while the system deactivates a region when there is a corner. The direction of the stroke end is used to determine which area to deactivate. The silhouette of the deactivated parts is rendered as broken lines.

Deactivation is provided in order to make the inside of an object accessible. The user can draw a contour and have it extrude on an internal surface in exactly the same way as done on an external surface (Fig. 5). The following sweep operation can also be used in conjunction with deactivation.

3.4 Sweep

After pressing the "Extrusion/Sweep" button, the user can also draw an open stroke specifying the sweep path. If a single contour is drawn in the first step,

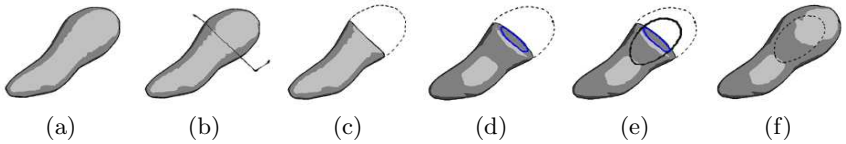

Fig. 4. An example of creating a hollow object: first, the user defines the desired cross-sectional plane by deactivating part of the object ((a-c)). Then, the user draws a contour on the plane (the blue loop in (d)). Finally, the user draws a extruding shape surrounding the contour, which we call "Loop Extrusion" (e). This creates a hollow object (f).

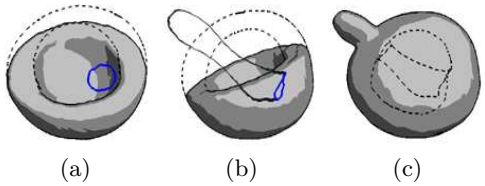

Fig. 5. A extrusion from an internal surface of an object using deactivation

both ends are checked to determine whether they are close to the projected contour. Unlike extrusion, the user can draw multiple contours to design tube-like shapes (Fig. 6).

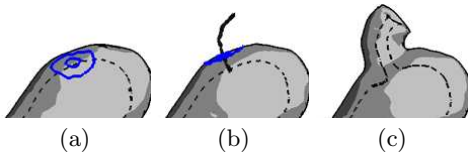

Fig. 6. Sweeping double contours: drawing contours on the surface of an object (a) and sweeping them (b) produces a tube (c).

3.5 Animation Assistance

In extrusion or sweep, the model must be rotated approximately 90 degrees after pressing the "Extrusion/Sweep" button to draw the last stroke. To automate this process, our system rotates the model after the "Extrusion/Sweep" button is pressed; the contours are then moved so that they are perpendicular to the screen (Fig. 7 (a-c)). This animation assistance is also performed after a Cut

operation, because it is likely that a contour will be drawn on the cut plane in the next step. When a model is cut, it is automatically rotated so that the cut plane is parallel to the screen (Fig. 7 (d-f)).

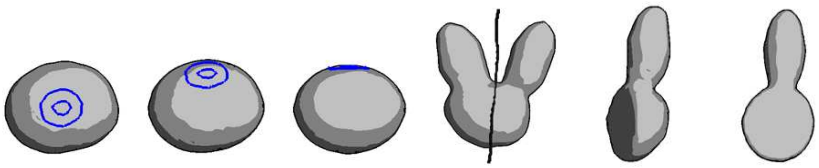

Fig. 7. Examples of animation assistance: as soon as the user presses the "Extrusion/Sweep" button, the model is rotated so that the contours are perpendicular to the screen (a-c). When the user cuts a model, the model is automatically rotated so that the cut plane is parallel to the screen (d-f).

4 Implementation

We use a standard binary volumetric representation. The examples shown in this paper require approximately 400^3 voxels. The volumetric data are polygonized using the Marching Cubes algorithm [9]. The polygonized surface is then smoothed [13] and displayed using a non-photorealistic rendering technique [8]. The silhouette lines of invisible or deactivated parts are rendered as broken lines.

The Create and Extrusion operations can be implemented using the algorithms described in the original Teddy system, converting the resulting polygonal model into a volumetric model and performing a CSG operation. In Extrusion, our system adds the additional geometry to the original model when an outward stroke is drawn and subtracts it when an inward stroke is drawn. Note that complex "sewing" of polygons is not necessary and no self-intersection will occur because of the volumetric data structure. Loop Extrusion applies the standard inward (subtract) extrusion in both directions. The Sweep operation in our system requires two-path CSG operations to add a new geometry to the original model. First, the sweep volume of the outermost contour is subtracted from the original model (Fig. 8 (a-c)). Then, the regions between the contours are swept and the sweep volume is added to the model (Fig. 8 (d)). This avoids the inner space being filled with the original geometry.

The volumetric representation significantly simplifies the implementation of the Cut operation and enables the change in topology. A binary 2D image is computed from the cutting stroke in the screen space to specify a "delete" region and a "remain" region. Both ends of the cutting stroke are extended until they intersect or reach the edges of the screen. Then, one of the separated regions is

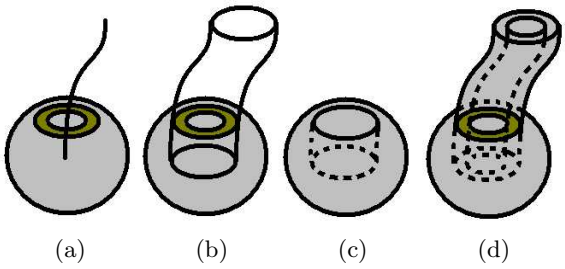

Fig. 8. Handling the sweep operation. The outmost contour is swept along the specified path (a,b) and extracted from the original model (c). Then, every contour is swept and added to the model.

set as the "delete" region (usually the region to the left of the stroke, following the original Teddy convention). Each voxel is then projected to the screen space to check whether it is in the deleted region; if so, the voxel is deleted. This process is significantly simpler than traversing the polygonized surface and remeshing it.

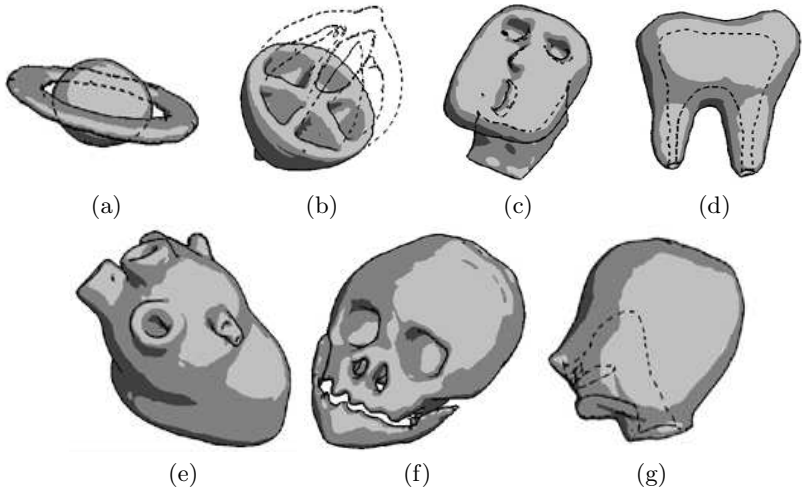

Fig. 9. Examples created using our system. (a-c) were created by novices, while (d-g) were created by an expert.

5 Results

We used a Dell Dimension 8200 computer that contained a Pentium 4 2-GHz processor and 512 MB of RAM. The graphics card was an NVIDIA GeForce3

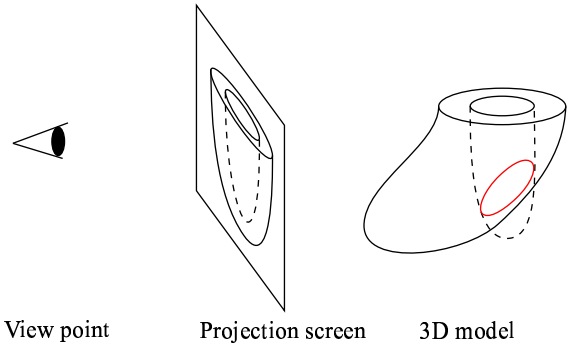

Fig. 10. An undesired effect caused by the lack of depth control. Since there is no depth information in the original model, the newly created cavity can pierce the wall.

Ti500 with 64 MB of memory. Users can create models interactively on this machine. We also used a display-integrated tablet as an input device, with which the user can edit an object more intuitively. However, some users found it difficult to rotate an object because they needed to press a button attached to the side of the pen and move the pen without touching the display.

Figure 9 shows some models created using our system. Fig. 9 (a-c) were created by novices within fifteen minutes of an introductory fifteen-minute tutorial; the others were created by an expert. Our observations confirmed that users could create models with internal structures quickly and easily. Nevertheless, one limitation also became clear. The users occasionally found the behavior of Extrusion unpredictable because there was no depth control. Specifically, when a user tried to create a cavity in an object, the hole sometimes penetrated the wall of the original model (Fig.10).

6 Conclusions and Future work

We presented a sketch-based modeling system for creating objects with internal structures. The underlying volumetric data structure simplifies the handling of a dynamically changing topology. The user can modify the topology easily in various ways, such as by cutting an object, forming a extrusion, specifying multiple contours with create or sweep operations, or specifying internal structures in conjunction with temporal deactivation. In addition, automatic rotation of the object frees the user from tedious manual labor.

Our system is designed for the rapid construction of coarse models and is not appropriate for precise modeling. Currently, it is difficult to modify shapes locally and we are exploring ways to add small details. As mentioned above, the absence of depth control causes difficulty. Finally, our current implementation

can produce only binary volumetric data and we plan to explore a new interface in which the user can define the internal *volumetric textures* of a model.

References

1. Bærentzen, J.A. and Christensen, N.J.: Volume Sculpting Using the Level-set Method. Proc. 2002 International Conference on Shape Modeling and Applications, (2002) 175–182.
2. Cutler, B., Dorsey, J., McMillian, L., Müller, M. and Jagnow, R.: A Procedural Approach to Authoring Solid Models. ACM Transactions on Graphics, **21**, 3, (2002) 302–311.
3. Ferley, E., Cani, M.P. and Gascuel, J.D.: Practical Volumetric Sculpting. The Visual Computer, **16**, 8, (2000) 469–480.
4. Galyean, T.A. and Hughes, J.F.: Sculpting: An Interactive Volumetric Modeling Technique. In Computer Graphics (Proc. SIGGRAPH 91), **25**, 4, (1991) 267–274.
5. Hua, J. and Qin, H.: Haptics-based Volumetric Modeling Using Dynamic Spline-based Implicit Functions. In Proc. 2002 IEEE Symposium on Volume Visualization and Graphics, (2002) 55–64.
6. Igarashi, T., Matsuoka, S. and Tanaka, H.: Teddy: A Sketching Interface for 3D Freeform Design. In Computer Graphics (Proc. SIGGRAPH 99), (1999) 409–416.
7. Karpenko, O., Hughes, J.F. and Raskar, R.: Free-form Sketching with Variational Implicit Surfaces. Computer Graphics Forum, **21**, 3, (2002) 585–594.
8. Lake, A., Marshall, C., Harris, M. and Blackstein, M.: Stylized Rendering Techniques for Scalable Real-Time 3D Animation. In Proc. Symposium on Non-Photorealistic Animation and Rendering (NPAR 2000), (2000) 13–20.
9. Lorensen, W.E. and Cline, H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In Computer Graphics (Proc. SIGGRAPH 87), **21**, 4, (1987) 163–169.
10. McDonnell, K.T. and Qin, H.: Dynamic Sculpting and Animation of Free-Form Subdivision Solids. The Visual Computer, **18**, 2, (2002) 81–96.
11. Perry, R.N. and Frisken, S.F.: Kizamu: A System for Sculpting Digital Characters. In Computer Graphics (Proc. SIGGRAPH 2001), (2001) 47–56.
12. Pugh, D.: Designing Solid Objects Using Interactive Sketch Interpretation. Computer Graphics (1992 Symposium on Interactive 3D Graphics), **25**, 2, (1992) 117–126.
13. Taubin, G.: A Signal Processing Approach to Fair Surface Design. In Computer Graphics (Proc. SIGGRAPH 95), (1995) 351–358.
14. Wang, S.W. and Kaufman, A.E.: Volume Sculpting. Computer Graphics (1995 Symposium on Interactive 3D Graphics), (1995) 151–156.
15. Zeleznik, R.C., Herndon, K.P. and Hughes, J.F.: SKETCH: An Interface for Sketching 3D Scenes. In Computer Graphics (Proc. SIGGRAPH 96), (1996) 163–170.