Progressive sketching with instant previewing

Wang, Kai; Zheng, Jianmin; Seah, Hock Soon

2019

Wang, K., Zheng, J., & Seah, H. S. (2019). Progressive sketching with instant previewing. *Computers & Graphics*, 81, 9–19. doi:10.1016/j.cag.2019.03.017

<https://hdl.handle.net/10356/144884>

<https://doi.org/10.1016/j.cag.2019.03.017>

© 2019 Elsevier Ltd. All rights reserved. This paper was published in *Computers & Graphics* and is made available with permission of Elsevier Ltd.

Downloaded on 26 Apr 2023 14:31:19 SGT

Progressive Sketching with Instant Previewing

Kai Wang, Jianmin Zheng*, Hock Soon Seah

School of Computer Science and Engineering, Nanyang Technological University, Singapore

Abstract

Sketch-based modeling uses freeform strokes as basic modeling metaphor and the modeling process typically involves two phases: sketching and sculpting. This paper presents a method to enhance the sketching phase to have the features of both sketching and sculpting. The key idea of the method is progressive sketching with instant previewing, which has the capacity of intuitively creating complex initial shapes that interpolate all the sketched strokes. The process lets the user iteratively sketch and meanwhile updates the reconstructed 3D shape instantly, which enables the user to be aware of the up-to-date shape. The underlying technique supporting this process is a novel surface construction algorithm that produces 3D triangular mesh models typically with only a single connected component and gradual shape changes during iterative sketching as well. To demonstrate the capability of the method, a sketching-based modeling prototype is also developed, which includes both sketching and sculpting functions. Experiments show that the proposed method can allow users to intuitively and flexibly create and edit 3D models even with complex topology, which is usually difficult for conventional sketch-based modeling systems.

Keywords: Sketch-based modeling, Sketching, Sculpting, Surface construction, Interface

1. Introduction

Shape modeling is a process of creating and manipulating geometric models and it is a fundamental task in geometric modeling and computer graphics. Many applications require digital 3D models, such as computer games, virtual/augmented reality, and scientific/medical visualization. There have been professional 3D modeling software tools available, such as SolidWorks and Autodesk's Maya and 3DS Max. These tools use the WIMP-style (Windows, Icon, Menu, Pointer) interfaces for the user to construct and manipulate 3D models, which is good for accurate modeling. However, the process and interaction are highly intensive and require skills that are not natural to designers or artists' habits. So far, 3D modeling is performed mostly by professionals. Unlike these modeling interfaces, sketch-based modeling was proposed to

use freeform curves as basic modeling metaphor and it allows the user to design 3D models in a way similar to the creation behavior of designers or artists in reality [1, 2, 3]. It is suitable for conceptual design, idea communication, shape retrieval, and shape modeling in real world and cyberworlds [4, 5, 6]. A sketch-based modeling system supporting multi-platform and allowing immersive 3D design through AR and VR has also been developed [7].

Sketch-based modeling has two basic processes: sketching and sculpting [4], which correspond to initial overall shape creation and subsequent shape editing, respectively. In the sketching phase, the user draws strokes as the contours of a desired shape and an initial 3D model is created after the sketching. In the sculpting phase, the user uses various sketches to perform carving, deformation, extrusion or smoothing to interactively refine the 3D model.

Based on the basic processes, sketch-based modeling systems can roughly be classified into two categories. The

*Corresponding author: J.Zheng (asjmzheng@ntu.edu.sg)

Figure 1: Progressive sketching with instant previewing.

first category is sculpting-dominant, which heavily relies on sculpting [8, 9]. Its sketching is rather simple. A simple sketching directly creates a very rough initial model, starting from which sculpting is applied to perform various shape editing. The advantage of this approach is that the sculpting process is intuitive and the user can see intermediate modeling results immediately after each sculpting process. The disadvantage is that the sculpting is very tedious and time-consuming when the target model is complex geometrically and/or topologically. Also note that in the sculpting process, the initial strokes drawn in the sketching phase, which represent the user’s intention about the shape of the target model, will generally be changed or discarded.

The second category is sketching-dominant, which mainly relies on sketching [10, 11, 12, 13]. It lets the user draw a set of feature lines or silhouettes of the target model, and then a 3D model is created after all the lines are drawn. The advantage of this approach is that the user can create an initial model is pretty much closer to the target shape and the sculpting process is used just to fine-tune the shape. The disadvantage is that there is no instant previewing of the intermediate shapes since the 3D model can be shown only after all the sketches are completed. As pointed out in [4], such sketching operation does not directly communicate about the shape that the user has in his/her mind. Without the help of visual for describing the 3D shape during drawings, sketching a 3D model

needs much skill, which restricts the techniques to trained and experienced people.

Considering the nature and advantages of sketching and sculpting, we propose to enhance sketch-based modeling by combining the modeling capability of sketching, and the intuitiveness and instant modeling of sculpting, which overcomes the disadvantages of sketching and sculpting mentioned above. This paper is an extension of the work we presented at International Conference of Cyberworlds 2018 [14]. The key idea is to realize progressive sketching with instant previewing, as illustrated in Figure 1. Specifically, in the sketching phase, the user is allowed to iteratively sketch. After each drawing, a new 3D model is created, which interpolates the new sketch as well as the old sketches drawn earlier. This empowers the sketching phase to create a complicated 3D model. Meanwhile the 3D models are progressively created and displayed immediately after each sketch, which makes the sketching process as intuitive as sculpting and makes the user aware of the 3D shape being sketched. To support this, a new surface construction algorithm is developed to build a 3D triangular mesh model with arbitrary topology from user-drawn sketches in real time. Comparing to existing construction algorithms, the new method has advantages such as avoiding unexpected shape changes during iterative sketching and producing meshes with a single connected component, which match the common expectation about progressive modeling.

The rest of the paper is organized as follows. Section 2 briefly reviews related work. Section 3 describes our sketching-based modeling prototype and Section 4 presents the underlying surface reconstruction algorithm that supports the proposed progressive sketching. The experiments are reported in Section 5 and Section 6 concludes the paper.

2. Related Work

Sketch-based modeling has been extensively studied. Many methods have been developed to create various shapes such as mesh surfaces [9, 15], implicit surfaces [16, 17] and subdivision surfaces [18]. There is rich literature on sketch-based modeling [1, 2, 19]. This section just briefly reviews those relevant to our work.

One class of sketching-based modeling is sculpting-dominant. It begins with a simple shape obtained from initial sketch(es) and then largely relies on various sculpting tools. One pioneer work is Teddy [8], with which the user draws a closed 2D contour, a simple oval-like object interpolating the contour is created, and then the user can use a set of sculpting tools to modify the initial shape until a desired shape is obtained. Typical sculpting tools include deformation [9, 20, 21, 22], merging [16, 23], extruding, cutting, smoothing [8, 9], feature sketching [9, 22], etc. Readers are referred to excellent survey papers [24, 19] for an overview of these sculpting functions. In general, if the target shape has complicated topology and geometry or is much different from the initially sketched shape, many sculpting operations are required, which makes the sculpting process tedious and time-consuming. Different from sculpting-dominant approaches, our work mainly focuses on techniques that help to create complicated initial shapes at the first stage (i.e., the sketching phase).

Another class of sketch-based modeling is sketching-dominant (i.e., focusing on sketching), which is similar to our work. The user sketches silhouettes or feature lines to depict the target shape. The algorithm interprets the strokes based on rules to map the 2D strokes into 3D space and constructs 3D models. For example, CrossSketch [10] uses the Cubic Corners to estimate the depth of intersection points of intersected 2D strokes, from which an open mesh surface can be constructed. The work [11] interprets the 2D curve between each pair of intersection

points as a 3D curve with minimum normal curvature. A closed mesh is formed by constructing surface patches bounded by the 3D curves. Masry and Lipson [12] presented a modeling system that constructs rigid 3D objects from straight and cursive lines. The Maximum weight Spanning Tree (MST) is used to determine the 3D depth of endpoints of straight lines and the curves in 3D space are calculated using the gradient steepest descent algorithm. Besides using rules, templates are also used to interpret the single view sketches [25, 26].

Moreover, it is more powerful to do sketching from multiple viewpoints. Ilovesketch [27] allows the user to sketch the 3D profile curves of the target model by using predefined gestures to change the viewpoints on a multi-touch display and the target model is constructed in a patch-by-patch manner similar to the methods in [11, 28]. Rivers et al. [29] designed a tool, with which the user sketches the silhouettes of a 3D object from three orthogonal views and then uses the visual hull to compute the target model. Note that these methods perform the construction only after all the sketches have been completed, which hinders providing instant hint of the 3D shape to the user during the sketching process. Different from these methods, our work provides instant previewing.

3. Sketching and Sculpting

This section describes an interface we have designed for our prototype, which provides both sketching and sculpting tools for the user to create and edit a 3D mesh model. The sketching tool is used for sketching an initial shape of the model progressively and the sculpting tools are used to edit or refine the created model.

3.1. Sketching tool

Note that one challenge in sketch-based modeling lies in the difficulty in specifying 3D position and orientation when drawing strokes into free 3D space, especially for novice users who are not familiar with virtual 3D space displayed on the 2D screen. Therefore our interface provides reference planes for the user to sketch. To simplify the discussion and implementation, this paper only considers three sets of orthogonal planes as reference planes. It is possible to extend the techniques and tool to planes with arbitrary orientation. The reference planes are sketch

Figure 2: An example of using the sketching tool. (a) Initially three basic planes are provided. (b) A closed curve is sketched on the blue plane. (c) The preview of the constructed shape. (d) A new cross section is added on the red plane and the previewed shape is rendered. (e) The green plane is translated and three more cross sections are sketched. (f) More cross sections are added to update the overall model. (g) The intersection curve between the green plane and the current model is rendered semi-transparently to provide reference for further sketches. (h) The final model.

planes, on which the user can sketch the cross sections of a 3D shape. *Cross section* curves are the intersection curves of the 3D shape with a plane. We choose the cross section as the input of the sketching tool for two considerations. First, cross sections are an intuitive and effective description of a 3D shape. Figuring out the shape from cross sections is familiar in traditional CAD design and medical applications. Comparing to silhouettes for shape description, cross sections may have fewer perception ambiguities. Second, a cross section is a planar curve, so it is relatively easy to sketch a cross section with the help of a plane.

With reference to Figure 2, initially, the $x = 0$, $y = 0$ and $z = 0$ planes serve as the three basic orthogonal planes (Figure 2(a)). The user sketches the cross section of a 3D model on any of these basic planes (Figure 2(b)). These three basic planes can be translated along x , y and z axes respectively to allow sketching of more contour curves (Figures 2(e,g)). This can result in several sets of parallel

cross sections to describe the shape of the model.

One important feature of our sketching tool is the progressive updating of the shape during sketching. That is, after each sketching, the model that interpolates all the up-to-date sketched strokes will be reconstructed and previewed instantly just as the sculpting process (Figures 2(c,f)), while most existing sketching-based systems create the model only after all the sketches are completed. In general, the sketching tool in the conventional systems uses an indirect way of communicating with a shape and it relies on the sense of human perception comparing to the sculpting tool [4]. We provide prompt previewing of the reconstruction result after each sketching, which can help the user be aware of the shape and his/her sketching. In addition, the current 3D shape also serves as a reference for the next sketch.

With the up-to-date model available and the current sketching plane, their intersection curve is computed and displayed semi-transparently to provide a reference for

the user to sketch on the sketching plane (Figures 2(g)). Also during the sketching process, all sketched cross sections are displayed and they can be modified through over-sketching or deletion. In addition, both the 3D model and the reference planes are rendered semi-transparently to avoid blocking the user’s view on existing sketches and the entire 3D space when providing sketching references.

Our tool is also able to automatically rotate the view to make the user-selected plane for sketching face him as much as possible. This is inspired by the observation in [27] that a good view for the user to sketch 2D strokes is one that has a large visible projection. With this function, frequent manual viewpoint changes that interrupt the user’s attention on sketching can be avoided.

It can be seen from Figure 2 that with the help of reference planes and the previewed shapes, more descriptions on the shape of a model can be sketched and the perception ambiguities can be reduced. This minimizes the unnecessary back and forth adjustment of sketches and speeds up the process of 3D modeling.

Note that our sketching cross sections on a reference plane that intersects the previewed shape looks similar to the sculpting process as in [30, 25], which creates a 3D model by iteratively oversketching its silhouette and deforming it. However, there are intrinsic differences between both processes. First, the cross sections that the user sketched earlier remain unchanged in our sketching process while they usually change with shape deformation in sculpting process. Second, the previewed shape in our process just provides a reference for subsequent sketches. It sets little limitation on the sketches. By contrast, in the sculpting process the sketches are restricted to the current 3D model.

3.2. Sculpting tools

After the user has sketched an initial shape of the model, a set of sculpting functions is provided for further editing and refinement. The sculpting functions include extrusion, tunneling (digging a hole), cutting, smoothing, and deformation, all of which are based on the user’s sketches. A stroke recognition mechanism is used to judge the user’s intention according to the sketches and then switch to the corresponding mode automatically. The sculpting tool accepts three types of strokes: a simple open stroke, a closed stroke, and a scratch-out stroke. The strokes are drawn with reference to the current model.

Figure 3: Sculpting examples. (a) A user sketched contour (yellow) and the generated initial model. (b) Tunneling operation. (c) Extrusion operation. (d) Smoothing operation. (e) Cutting operation.

Based on the types of strokes, reference planes will be automatically generated for subsequent drawing.

- When a closed stroke is sketched on the surface of the model, it triggers an extrusion or tunneling as in [8, 9]. Different from [8, 9], our tool automatically generates a reference plane orthogonal to the closed curve. The plane provides a reference in 3D space for the user to draw the silhouette curve of the extruded/dug part (Figure 3(c)). The orientation of the plane can be further adjusted by performing rota-

Figure 4: A deformation example. (a) The blue curve is the handle curve sketched on the initial model. The yellow, red and green dots denote the handle, ROI and static vertices respectively. (b) The automatically generated base plane (blue) and projection plane (green). (c) The target curve (green) drawn on the base plane. (d) To create a non-planar target curve, a curve (red) is drawn on the projection plane as its projection. (e) The updated non-planar target curve (green). (f) The deformed model.

tion about one of two axes: the line passing through the center of the closed stroke and orthogonal to the stroke as much as possible or the line connecting the first drawn point of the closed stroke and the farthest point to it on the stroke. The reference plane helps to determine the position of the silhouette curve and gives flexibility to adjust its direction. When the silhouette curve intersects with the back of the surface, the operation becomes tunneling and a hole is dug (Figure 3(b)).

- When the user scratches a part on the surface back and forth, it is regarded as a smoothing operation and the corresponding local part will be smoothed (Figure 3(d)).
- When a simple open stroke is sketched, the operation may be cutting or deformation, depending on whether the stroke passes across the model in the image space or not. If it is a cutting stroke, a following stroke will be required to indicate which part to cut off (Figure 3(e)), or a following stroke will be drawn in an empty region to indicate that the model is cut into two parts in order to generate multiple-components.
- If a simple open stroke does not pass across the whole model in the image space, the stroke is mapped to the surface and serves as a handle curve for deformation. The deformation of the surface is driven by the modification of the handle curve. Curves have been shown to be more effective in manipulating the shape deformation [9]. See Figure 4

for an example.

The handle curve is represented by a polyline consisting of vertices on the surface, called the handle vertices (Figure 4(a)). Based on the stroke, two reference planes are generated for the user to sketch the modified shape of the handle curve: a base plane that passes as many as possible handle vertices for sketching the basic shape of the target curve and a projection plane that is orthogonal to the base plane for drawing the projection of the target curve (see Figure 4(b)). In general, the base plane is sufficient for sketching the target shape of the handle curve. However, if the target handle curve is desired to be non-planar, the projection plane is used to sketch the projection of the target curve. The two planar curves on the base plane and the projection plane are combined to determine a non-planar handle curve (Figure 4(c) - Figure 4(e) using an algorithm proposed in [31]. In addition, the two planes can also be rotated to provide more flexibility. In this way, the user can generate complex 3D target curves.

Further, the user needs to specify the vertices of the mesh within a region of interest (ROI), which will move during the deformation. This is done by three methods. A simple method is to specify a threshold and then to select those vertices on the mesh, whose Euclidean distances to the handles are shorter than a threshold. A second method is to let the user draw a circle on the 2D screen. The ROI vertices are the ones whose projections onto the screen are inside the circle. To have more control in selecting the ROI

vertices for complex models, a third method is to use a brush. The vertices brushed are chosen as the ROI vertices. After this, the deformation is performed, which pulls the handle curve to the target location and adjusts the ROI vertices accordingly. The rest of the vertices remain unchanged and are called static vertices (see Figure 4(a)).

3.3. Generation of reference planes for sculpting

One problem in the sculpting phase is how to automatically generate reference planes, which have appropriate position and direction for the user to further sketch, once a stroke is drawn with reference to the current model to trigger a deformation, extrusion or tunneling. For a deformation stroke, we propose to let the base plane pass through as many points of the handle curve as possible. Hence the plane can be computed by the least squares fitting. In particular, assume the plane equation is

$$Ax + By + Cz + D = 0,$$

where coefficients A, B, C and D are to be determined. Our goal is to find A, B, C and D as the solution to the following constrained optimization problem:

$$\begin{aligned} \min_{(A,B,C,D)} \sum_{P_i \in HC} (Ax_i + By_i + Cz_i + D)^2 \\ \text{subject to } A^2 + B^2 + C^2 = 1 \end{aligned} \quad (1)$$

where HC is the set of points $P_i = (x_i, y_i, z_i)$ on the handle curve. The optimization problem can be solved using the eigenvalue minimization [32].

The projection plane is computed from the base plane. First, we project the starting point of the handle curve onto the base plane. Second, we choose another projection point of the handle curve on the base plane, which is farthest to the projection of the starting point. Connecting the two projection points forms a line. Finally, we rotate the base plane about the line by 90 degrees, which gives the projection plane.

For an extrusion or tunneling stroke, we compute the reference plane in a way similar to computing the projection plane. We first compute an initial least squares plane for the stroke. Then we compute a line on the plane as the rotation axis, which connects two projection points of the stroke. Finally the desired reference plane is obtained by rotating the initially obtained plane about the axis by 90 degrees.

Note that the above methods just give a suggestive orientation of the reference plane. If the user is not satisfied with it, he can rotate the plane about a line which is constructed similarly as in computing the projection plane.

4. Surface Construction

This section presents a surface construction algorithm that is specifically designed to support and implement the proposed progressive sketching with instant previewing. The input to our surface construction algorithm is a set of cross sections $\{cs_i \mid i = 1, \dots, m\}$ which are sketched by the user and lie on a set of orthogonal planes in 3D space $\mathbb{R}^3$. Without loss of generality, each plane is defined by $x = d_i$, $y = h_j$ or $z = g_k$ where $d_i, h_j, g_k \in \mathbb{R}$. Each cross section is represented by a polyline and does not self-intersect or intersect any other cross section curves lying on the same plane. Moreover, for two cross section curves cs_1 and cs_2 lying on two orthogonal planes P_1 and P_2 respectively, the intersection point set of curve cs_1 and plane P_2 should be the same as that of curve cs_2 and plane P_1 . These constraints on the cross section curves are imposed to avoid singular or ambiguous situations in defining a valid object [33]. Otherwise, some repairing procedures are needed [34]. The output is a closed 2-manifold triangular mesh interpolating the sketched cross sections. Moreover, to make the sketching process intuitive and progressive modeling natural, the construction should have the following properties:

- The shape change of the reconstructed models with the adding of new sketches should be gradual. This makes the change of the shape predictable, which meets the common expectation of progressive modeling.
- The construction result should be independent of the user's sketching order. That is, given a set of curves drawn with different orders, the construction results should be the same (see Figure 5).
- The constructed model should avoid having multiple unconnected components. This is consistent with human's shape perception.

There have been many 3D surface reconstruction algorithms. Early work focused on the construction from

Figure 5: Three rows show different orders of sketching the same set of strokes. It is expected that the construction result is independent of the order.

parallel cross sections [35, 36]. Later, the construction from non-parallel cross sections has also been developed [37, 38]. For example, Liu et al. [38] proposed a projection method to handle cross sections on non-parallel planes. The method first partitions the 3D space into zones bounded by the planes that the cross sections lie on. Then within each zone, the Medial Axis (MA) planes are calculated, and the cross sections are projected onto the MA planes. Via the projections the cross sections are connected to form an initial mesh, which is further refined through refinement and smoothing. Note that this method relies on the construction of MA planes of zones. Each time a new cross section on a new plane is sketched, the new plane may cause splitting of some existing zones and then the MA planes of the split zones should be re-calculated, which will lead to the re-calculation of projections of all cross sections in these zones and thus the change of the surface. This sometimes results in unexpected (or unnatural) shape change in our progressive sketching especially when the MA structure changes.

To overcome the above problem, our idea is to extrude the cross sections to create 3D solids and use a union operation to build an object whose boundary surface interpolates the cross sections, rather than using the projections on the MA planes. This step is mainly to build the topology of the model. The geometry of the model is improved by a constrained smoothing procedure. The algorithm roughly consists of the following steps.

- (1) **Space partition:** The planes where the cross sections lie are used to partition the space into zones. Each zone is a cuboid bounded by $x = x_i$, $x = x_{i+1}$, $y = y_j$, $y = y_{j+1}$, $z = z_k$ and $z = z_{k+1}$ where $x_i, x_{i+1}, y_j, y_{j+1}, z_k$ and z_{k+1} are some scalars. If a cross section curve bounds a region that lies in more than one zone, we split the region by the edges of the zones so that the each sub-region is entirely in a face of a zone. We also call the boundary of each sub-region a cross section curve without causing ambiguity. We classify a zone into one of the three types based on how many faces of the zone contain a cross section curve: an *empty zone* that has no face containing a cross section; an *end zone* that has only one face containing a cross section; and a *body zone* that has two or more faces containing a cross section.
- (2) **Extrusion in zones:** In each zone we construct a rough 3D solid for each cross section on the faces of the zone. Particularly, for a body zone, we extrude each cross section along the normal of the face it lies on, which yields an extruded model. For an end zone, we not only extrude the cross section along the normal of the face it lies on but also extend the extrusion till a body zone or a zone without a neighboring zone in the normal direction is obtained. The purpose of performing extension is to reduce the chance of occurrence of isolated components in the final construction. For an empty zone, it does not contain cross sections and is thus ignored. All the surfaces of

the extruded solids are further triangulated into triangular meshes.

- (3) **Merging extruded models:** Once all the extruded solids are built in each zone, we compute their union using the algorithm of [39]. The algorithm is robust and fast for our application since all the input polyhedra are extruded solids. The boundary surface of the union produces an initial triangular mesh for construction. Based on the construction of extrusion and union, it is obvious that the initial triangular mesh interpolates the cross sections.
- (4) **Constrained smoothing:** The generated triangular mesh provides an initial overall model that nicely connects all the cross sections, but it is generally of low quality in terms of geometry (see Figures 6(g) and 7(e)). To improve the quality, we perform a smoothing process. Due to its simplicity and effectiveness, the Laplacian-based fairing method is chosen to smooth the mesh vertices except the vertices on the input cross sections. This constrained approach assures proper interpolation of cross sections and avoids shrinking. If the constructed mesh is too coarse, local refinement is performed [38]. Note that the constrained smoothing not only improves the quality of the constructed mesh but also makes the final result insensitive to the geometry of extruded models in each zone.

Below we explain the details of performing extrusion in body zones and end zones.

4.1. Extrusion in a body zone

Consider a body zone zn_i . It has at least two faces that contain at least a cross section. If a cross section cs_p on face f_{ij} of zn_i does not intersect any other cross sections in zn_i , we extrude cs_p towards the inside of zn_i along the normal direction of face f_{ij} by a distance h_p , generating a prism cl_p . Denote the dimension of zn_i along this normal direction by h_{ij} . In case h_{ij} is infinity, we set it to be a default positive value. Then the extrusion distance h_p is defined by

$$h_p = \delta h_{ij} \quad (2)$$

where δ is a positive number in $(0, 1)$ to make sure that the top face of the prism cl_p is close to but does not reach the opposite face of f_{ij} . The top and bottom faces of the

prism are polygons. They are triangulated into a set of triangles. The side surface of the prism consists of a set of rectangles. Each rectangle is split into two triangles. In this way, the surface of prism cl_p becomes a triangular mesh that is contained in zone zn_i and interpolates cross section cs_p . Figure 6 illustrates such a construction of prisms.

If a cross section cs_p on face f_{ij} intersects another cross section cs_q on a neighboring face f_{ik} of f_{ij} in zone zn_i as shown in Figure 7(a), constructing prism cl_p in the above approach will make one side face of cl_p lie on f_{ik} or overlap cross section cs_q . As a result, the surface of the union of the prisms may not interpolate cs_q on f_{ik} . Therefore in this case, we need to modify the prism. Specifically, we first generate a prism cl_p^{ini} from cs_p by extrusion as before. Denote the top polygon of prism cl_p by cl_p^t . Then we move those points of cl_p^t , which lie on face f_{ik} , towards the interior of cl_p^t by a small distance so that the perturbed extrusion cl_p does not have faces lying on f_{ik} . After that, we triangulate the surfaces of cl_p in a similar way. Note that the perturbation does not affect the union calculation since the algorithm accepts general polyhedra as input, and the surface of the perturbed extrusion still interpolates cs_p . Figure 7 illustrates the process for the situation with intersecting cross sections.

4.2. Extrusion in an end zone

Consider an end zone zn_i . It has only one face containing cross section curve(s). Denote this face by f_{ij} . For a cross section cs_p on f_{ij} , we can extrude it in zone zn_i to generate a prism. This prism will not be connected to other extrusions except the one generated also from cs_p in the other zone that shares face f_{ij} with zn_i . As a result, this prism may become an isolated surface component. On the other hand, in our progressive sketching, the user usually adds new sketches near the up-to-date shape and therefore the construction result is expected to update the shape without leading to multiple components. Note that many end zones result from splitting a body zone due to the insertion of a new plane. This observation suggests that if we extend the extrusion, the possibility for the extrusion to become an isolated component will be reduced.

Specifically, suppose that the opposite face of f_{ij} in zone zn_i is f_{ik} . If f_{ik} is not at infinity, we extrude cs_p across face f_{ik} into the neighboring zone zn_l that shares face f_{ik} with zone zn_i . We continue this extension process

Figure 6: Surface construction from non-intersecting cross sections in *body* zones. (a) The input cross sections; (b) Partition result, where the numbers represent the zone indices; (c)-(e) Prisms generated from the two cross sections in Zone 1; (f) The unions of prisms; (g) The final construction result after refinement and smoothing.

until a body zone is reached or further extension to a new zone is impossible. The extrusion in the last zone is then done just as in Section 4.1. Similarly, the surface of the extension is also triangulated.

4.3. Discussion

Note that during progressive sketching, some zones may be split, which causes the change of MA planes in the method of [38]. The change of MA planes is in general complicated and sensitive. This may lead to unexpected or unnatural change of reconstructed shape. In contrast, the user’s sketches represent the cross sections of the shape to be modeled, so the extrusion of the cross sections roughly reflects the shape and therefore our extrusion and union based approach will give a more natural update of the model in the progressive sketching.

5. Experiments

Our sketch-based modeling is implemented using C++. The experiments are carried out on an Intel Pentium 4

3.6GHz machine with 1G RAM. Both the surface construction and other sculpting functions run in an interactive rate, which meets the requirement for a sketch-based modeling system.

First, we compare the reconstruction with and without using the extension strategy. Figure 8 is an example showing the extension process. The result without using the extension is given in Figure 8(f), which has four unconnected components. It can be seen that by extending extrusions in end zones, the possibility for them to be connected to other extrusions greatly increases and hence isolated components are unlikely to occur.

Second, we compare our method with [38] in terms of gradual change of shapes. Figures 9(a)-(c) show the process of the MA-based approach [38]. When the third stroke is added, the originally connected component is broken, which is unlikely the user’s intention. With our method as shown in Figures 9(d) and (e), the change of the shapes looks more gradual and natural.

Third, we invited ten novice users for a user study. We trained them for about 20 minutes and then asked them

Figure 7: Surface construction from intersecting cross sections in *body* zones. (a) The input cross sections; (b)-(c) Prisms are generated from the two cross sections and perturbed; (d) The union of prisms in a zone; (e) The initial construction result; (f) The final construction result after refinement and smoothing.

Figure 8: Surface construction in *end* and *body* zones. (a) Input cross sections; (b) Space partition; (c) The extended prisms for *end* zones. The blue faces are those that may cause isolated surface components; (d) The unions of prisms in all zones; (e) Final surface construction result using the extension approach; (f) Construction result without using the extension.

Figure 9: Comparison of our algorithm and [38]. (a) Two cross sections are sketched first (top) and then the third one is added (bottom); (b) The initial construction results by [38], where the Medial Axis planes in a zone are shown in blue; (c) The final construction results by [38]; (d) The initial construction results by our algorithm; (e) The final construction results by our algorithm.

to create some freeform shape models. It was observed that most users tended to use the sketching tool to iteratively sketch and construct the 3D models in most of the time and the sculpting tools were taken just to do some detailed modifications. Very few preferred to build a model by sculpting from a very rough shape. This is consistent with the observation in [4] that the sketching technique is more familiar to most people than the sculpting technique that requires tedious manual work. Figure 10 shows some models created using the sketching tool only. Figure 11 shows two models that were created using both sketching and sculpting tools. .

We learned from the feedback that with the help of the reference planes, users can get a direct “touch” of the 3D

space and they only need to focus on the sketching of the curves for defining the shape of the 3D model. The up-to-date shape together with the existing sketches lets them get a clear idea of the 3D model in each design stage and also gives hints for the next sketches. This can overcome the difficulties on confirming the position and orientation of the 3D curves corresponding to the 2D sketches and imagining the sketched 3D shape.

6. Conclusion

We have described a novel sketching method that allows the user to progressively create and edit 3D freeform shape. The main technical contributions of the paper lie

Figure 10: Examples created using sketching.

in three aspects. First, we propose to enhance the sketching process in sketch-based modeling by instantly updating the shape during iterative sketching of cross section curves in the way of sculpting, which combines the advantages of both sketching and sculpting. This makes the sketching phase more powerful and meanwhile intuitive and simple. Second, a new surface construction algorithm is developed to effectively support the proposed sketching process. Compared to previous work, the algorithm

has several advantages: (1) the constructed 3D models are updated gradually during iterative sketching; (2) the final sketched model is independent of sketching order; and (3) the constructed 3D models likely have a single connected component. Third, we develop a sketch-based modeling prototype that provides both sketching and sculpting tools for shape modeling. The sketching tool is based on progressive sketching with instant previewing.

While the current implementation of the progressive

Figure 11: Examples created using sketching and sculpting.

sketching and the sketch-based tool is based on three sets of orthogonal planes as references, in future we will extend the technique and tool to arbitrary reference planes. In addition, there is need to develop a more effective constrained smoothing procedure to improve the geometry of the constructed models.

Acknowledgments

This work is supported by MoE Tier-2 Grant (2017-T2-1-076) of Singapore.

References

- [1] Jorge, JA, Samavati, F, editors. Sketch-based Interfaces and Modeling. Springer; 2011. ISBN 978-1-84882-811-7. doi:10.1007/978-1-84882-812-4.
- [2] Cordier, F, Singh, K, Gingold, Y, Cani, MP. Sketch-based modeling. In: SIGGRAPH ASIA 2016 Courses. SA '16; ACM; 2016,.
- [3] Li, C, Pan, H, Liu, Y, Tong, X, Sheffer, A, Wang, W. Bendsketch: Modeling freeform surfaces through 2d sketching. ACM Trans Graph 2017;36(4).
- [4] Cani, M, Igarashi, T, Wyvill, G. Interactive shape

- design. *Synthesis Lectures on Computer Graphics and Animation* 2008;3(1):1–78.
- [5] Li, B, Lu, Y, Godil, A, Schreck, T, Bustos, B, Ferreira, A, et al. A comparison of methods for sketch-based 3d shape retrieval. *Computer Vision and Image Understanding* 2014;119:57–80.
 - [6] Zhang, X, Goesele, M, Sourin, A. Haptic interaction with a polygon mesh reconstructed from images. In: *2016 International Conference on Cyberworlds*. 2016, p. 49–56.
 - [7] Gravity Sketch. <https://www.gravitysketch.com/>; 2019. Accessed: 2019-03-06.
 - [8] Igarashi, T, Matsuoka, S, Tanaka, H. Teddy: a sketching interface for 3d freeform design. In: *SIGGRAPH '99*. 1999, p. 409–416.
 - [9] Nealen, A, Igarashi, T, Sorkine, O, Alexa, M. Fibermesh: designing freeform surfaces with 3d curves. *ACM Transactions on Graphics* 2007;26(3).
 - [10] Andre, A, Saito, S, Nakajima, M. Crosssketch: freeform surface modeling with details. In: *SBIM'07*. 2007, p. 45–52.
 - [11] Koel, D, Pablo, D, Gopi, M. Sketching free-form surfaces using network of curves. In: *SBIM'05*. 2005,.
 - [12] Masry, M, Lipson, H. A sketch-based interface for iterative design and analysis of 3d objects. In: *ACM SIGGRAPH 2007 courses*. 2007, p. 31.
 - [13] Orbay, G, Kara, L. Sketch-based surface design using malleable curve networks. *Computers & Graphics* 2012;36(8):916 – 929.
 - [14] Wang, K, Zheng, J, Seah, HS. Enhancing sketching and sculpting for shape modeling. In: *International Conference on CyberWorlds, Singapore, October 3-5, 2018*. 2018,.
 - [15] Takayama, K, Panozzo, D, Sorkine-Hornung, A, Sorkine-Hornung, O. Sketch-based generation and editing of quad meshes. *ACM Transactions on Graphics* 2013;32(4).
 - [16] Schmidt, R, Wyvill, B, Sousa, M, Jorge, J. Shapeshop: sketch-based solid modeling with blob-trees. In: *SBIM'05*. 2005, p. 53–62.
 - [17] Brazil, E, Macedo, T, Sousa, M, de Figueiredo, L, Velho, L. Sketching variational hermite-rbf implicits. In: *SBIM'10*. 2010, p. 1–8.
 - [18] Nasri, A, Karam, W, Samavati, F. Sketch-based subdivision models. In: *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*. 2009, p. 53–60.
 - [19] Ding, C, Liu, L. A survey of sketch based modeling systems. *Front Comput Sci* 2016;10(6):985–999.
 - [20] Sugihara, M, Wyvill, B, Schmidt, R. WarpCurves: A tool for explicit manipulation of implicit surfaces. *Computers & Graphics* 2010;34(3):282–291.
 - [21] Zimmermann, J, Nealen, A, Alexa, M. Silsketch: automated sketch-based editing of surface meshes. In: *Proceedings of the 4th Eurographics Workshop on Sketch-Based Interfaces and Modeling*. 2007, p. 23–30.
 - [22] Paiva, A, Amorim, R, Velho, L, Sousa, M. Sketch-based adaptive mesh augmentation using stellar operators. In: *SIBGRAPI '11: Proceedings of the 24th SIBGRAPI Conference on Graphics, Patterns and Images*. 2011, p. 78–85.
 - [23] Schmidt, R, Singh, K. Sketch-based procedural surface modeling and compositing using surface trees. *Computer Graphics Forum* 2008;27(2):321–330.
 - [24] Olsen, L, Samavati, F, Sousa, M, Jorge, J. Sketch-based modeling: A survey. *Computers & Graphics* 2009;33(1):85–103.
 - [25] Kraevoy, V, Sheffer, A, van de Panne, M. Modeling from contour drawings. In: *SBIM'09*. 2009, p. 37–44.
 - [26] Xu, K, Chen, K, Fu, H, Sun, W, Hu, S. Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics* 2013;32(4).

- [27] Bae, S, Balakrishnan, R, Karan, S. Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In: Proceedings of the 21st ACM Symposium on User Interface Software and Technology. 2008, p. 151–160.
- [28] Abbasinejad, F, Joshi, P, Amenta, N. Surface patches from unorganized space curves. *Computer Graphics Forum* 2011;30(5):1379–1387.
- [29] Rivers, A, Durand, F, Igarashi, T. 3d modeling with silhouettes. *ACM Transactions on Graphics* 2010;29(4).
- [30] Nealen, A, Sorkine, O, Alexa, M, Cohen-Or, D. A sketch-based interface for detail-preserving mesh editing. *ACM Transactions on Graphics* 2005;24(3):1142–1147.
- [31] Cohen, J, Markosian, L, Zeleznik, R, Hughes, J, Barzel, R. An interface for sketching 3d curves. In: *I3D '99: Proceedings of the 1999 Symposium on Interactive 3D graphics*. 1999, p. 17–21.
- [32] Schomaker, V, Waser, J, Marsh, R, Bergman, G. To fit a plane or a line to a set of points by least squares. *Acta Crystallographica* 1959;12(8):600–604.
- [33] Wang, K, Zheng, J, Seah, HS. Reference plane assisted sketching interface for 3d freeform shape design. In: *2010 International Conference on Cyber-Worlds*, Singapore. IEEE Computer Society. ISBN 978-0-7695-4215-7; 2010, p. 105–112.
- [34] Huang, ZY, Holloway, M, Carr, N, Ju, T. Repairing inconsistent curve networks on non-parallel cross-sections. *Comput Graph Forum* 2018;37(2):25–35.
- [35] Keppel, E. Approximating complex surfaces by triangulation of contour lines. *IBM J Res Dev* 1975;19(1):2–11.
- [36] Boissonnat, J. Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing* 1988;44(1):1–29.
- [37] Boissonnat, J, Memari, P. Shape reconstruction from unorganized cross-sections. In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, Barcelona, Spain, July 4-6, 2007. 2007, p. 89–98.
- [38] Liu, L, Bajaj, C, Deasy, J, Low, D, Ju, T. Surface reconstruction from non-parallel curve networks. *Computer Graphics Forum* 2008;27(2):155–163.
- [39] Laidlaw, D, Trumbore, W, Hughes, J. Constructive solid geometry for polyhedral objects. *Computer Graphics (Proc SIGGRAPH 1986)* 1986;20(4):161–170.