Freeform extrusion by sketched input

Charlie C.L.WANG*, Matthew M.F.YUEN

*E-mail: wangcl@ust.hk or mewangcl@hotmail.com; Tel: (852) 23588095; Fax: (852) 23359298
Department of Mechanical Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Abstract

This paper presents a sketch based mesh extrusion method, which is useful for intuitive, efficient geometric modeling of freeform polygonal objects. With our method, the user can extrude either a closed or open surface from the surface or the boundary of a given mesh by sketched input. Thus, compared to earlier methods, our method provides more flexibility for solving surface modeling problems. Our mesh extrusion algorithm consists of four steps: 1) create a base curve; 2) project extruding strokes; 3) sweep a base curve; and 4) sew adjacent curves. At the end of this paper, examples of our mesh extrusion implementation are shown to demonstrate its functionality. Our freeform extrusion has been implemented as a part of our sketch based modeling system, and the mesh extrusion can be completed in real-time on a standard PC.

Keywords: computational geometry and object modeling; interaction styles; computer-aided design.

1. Introduction

This paper describes the development of an intuitive and efficient geometric modeling method for designing freeform polygonal models. In currently available commercial geometric modeling software, there are different kinds of modeling tools. But all of these tools are for precise design. They are not efficient or intuitive enough for conceptual design. It is important to develop an efficient and intuitive tool for conceptual design. Studies show that most designers nowadays still prefer to express their creative design ideas through 2D sketches. Thus, a sketched input method is needed.

Our mesh extrusion method is best illustrated by an example shown in Fig. 1. In this example, the given initial model is a closed mesh surface, our approach allows a designer to sketch two or three strokes to extrude a surface from the given mesh – a closed stroke on the surface of the given mesh, and one or two 2D strokes depicting the silhouette of the extruded surface. Firstly, the user draws a closed stroke on the object surface. A surface curve is obtained by projecting the closed stroke onto the surface of the given mesh – called a base curve. After that, the user rotates the model to bring the closed stroke sideways and draws silhouette lines to extrude the surface (Fig. 1a). A sweep operation is applied to construct the 3D shape by moving the surface curve along the skeleton of the silhouette. Fig. 1b shows the result of the extrusion, and Fig. 1c shows the mesh representation of the extrusion result. The direction of the extrusion is perpendicular to the object surface, but not parallel to the screen. However, since the problem of getting a 3D shape of a surface from its contour is a problem with insufficient constraints, our algorithm uses some assumptions as additional constraints to solve the problem. For example, the cross-section shape of the extruded surface is assumed to be similar to the shape of the base curve; and the end shape of the open surface extruding is also assumed to be similar to the shape of the base curve. The whole operation only consists of two or three strokes (two strokes to construct a closed surface – e.g., the top row in Fig. 1, and three strokes to construct a open surface – e.g., the bottom row in Fig. 1). It is very intuitive and efficient. Our approach uses the polygonal mesh to represent a surface, which is widely used as a fundamental representation for geometric modeling.

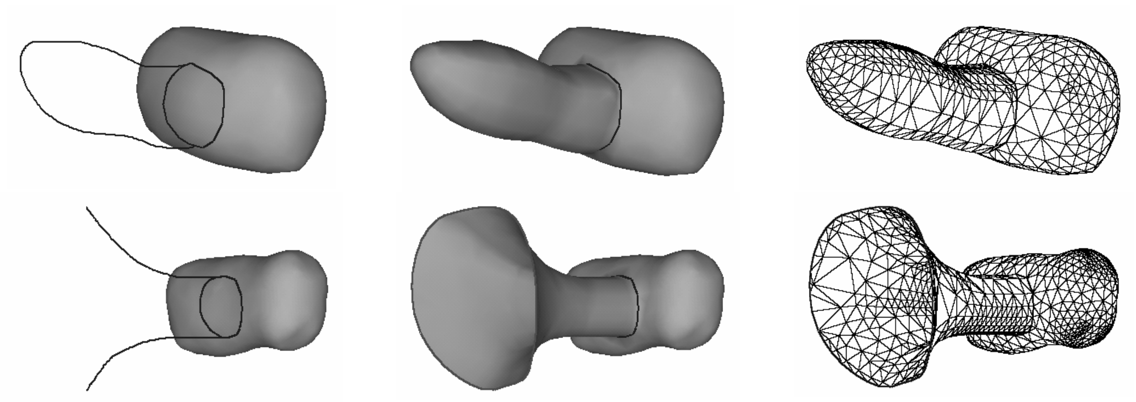
(a) sketched input **(b) extrusion result** **(c) mesh representation**

Fig. 1. Freeform extrusion

In this paper, we propose a mesh extrusion algorithm that consists of four steps: 1) create a base curve; 2) project extruding strokes; 3) sweep a base curve; and 4) sew adjacent curves. After the mesh extrusion, a generic mesh optimization scheme [1] is implemented to optimize the constructed surface with respect to prescribed criteria. Our extrusion approach is closely related to the extrusion method in the Teddy system [2]. But by our method, the user can extrude both a closed and open surface from either the surface or the boundary of an initial mesh; the Teddy system can only extrude a closed surface from the surface of an initial mesh. Therefore, our method provides more flexibility in dealing with surface modeling problems. In other words, the limitation of spherical topology is overcome.

2. Related Work

Much research focused on mesh surface construction [2, 3, 4, 5, 6, 7, 8, 9]. Meyers et al.'s work [3] was concerned with the problem of reconstructing the surfaces of three-dimensional objects with a collection of planar contours representing cross-sections through the objects. Their approach was related to Keppel's work [4]. Hoppe et al. [5] presented a general method for the automatic reconstruction of accurate, concise and piecewise smooth surface models from scattered range data. The SKETCH system [6] can rapidly construct an approximate shape via direct mark based interaction. The Teddy system [2] constructed a rounded freeform mesh model by finding the chordal axis of user input 2D closed stroke to build a smooth surface around the axis, and the Teddy system can extrude closed surface from the surface of a given mesh. Our proposed method, which can extrude either a closed or an open surface from either the surface or the boundary of the given initial mesh, provides more flexibility for surface modeling problems. Implicit surface modeling [7, 8, 9] is another approach for mesh construction. The user specifies the skeleton of the intended model and the system constructs smooth, natural-looking surfaces around it. By extending implicit surface modeling techniques, Markosian et al. [9] presented a particle-based freeform surface modeling approach. In their system, a user interactively guides the particles, called skin, to grow over a given collection of polyhedral elements (or skeletons), yielding a smooth surface (through subdivision) that approximates the underlying skeletal shapes. In our mesh extrusion algorithm, we use the approach of Meyers et al. [3] to construct a mesh surface between two neighboring base curves.

Suzuki et al. [10] presented a 3D mesh-dragging method for intuitive, efficient geometric modeling of free-form polygonal models. With their method, the user can drag a part of a triangular mesh and change its position and orientation. Their method was based on an adaptive remeshing procedure. Similarly, our interactive mesh extrusion also proposes an intuitive, efficient geometric modeling tool for free-form polygonal models. Some other interactive modeling methods are related to the multi-resolution presentation of models: Zorin et al. [11] built a scalable interactive multi-resolution editing system based on mesh refinement and coarsification algorithms; and Khodakovsky and Schroder [12] developed an algorithm based on Zorin's approach that can modify the fine level shape of a surface.

Several algorithms have been formulated to simplify surfaces and to solve the more general problem of multi-resolution modeling. Schroeder et al. [13] described an algorithm that we term vertex decimation. Their method iteratively selects a vertex for removal, removes all adjacent faces and re-triangulates the resulting hole. Garland and Heckbert [14] developed a surface simplification algorithm that can rapidly produce high quality approximations of polygonal models. The algorithm uses iterative contractions of vertex pairs to simplify models and maintain surface error approximations using quadric matrices. It can facilitate good approximations, both visually and geometrically. But in their algorithm, the element shape qualities are not respected. Hoppe et al. [15] presented a method for solving the mesh optimization problem by defining an energy function that explicitly models the competing requirements of conciseness of representation and fidelity to the data. Their algorithm is based on obtaining the optimal solution of energy functions. It is a time-consuming iteration scheme, and the cloudy points are required as references. Frey and Borouchaki [16] presented a surface mesh optimization method that is suitable for obtaining a geometric finite element mesh, given an initial arbitrary surface triangulation. The first step of their method involves constructing a G^1 continuous geometric support, associated with the initial surface triangulation, which represents an adequate approximation of the underlying surface geometry. The initial triangulation is then optimized with respect to this geometry as well as the element shape. Their procedure of mesh optimization is also a very time consuming iteration scheme, but not a real time mesh optimization technique, and the reference parametric surface is required. Since our extrusion approach is a part of an interactive design system, we should choose real-time algorithms [1, 17] that can optimize an arbitrary triangular mesh surface with respect to a prescribed criteria to obtain a unit surface mesh (i.e., where all edges have a normalized length close to the optimal value). Here, we use the algorithm presented in [1].

3. Mesh Extrusion

Our extrusion tool allows a designer to sketch two or three strokes to extrude a surface from the given mesh Φ – a closed stroke on the surface of a given mesh, and one or two 2D strokes depicting the silhouette of the extruded surface. The algorithm consists of four steps: 1) create a base curve; 2) project extruding strokes; 3) sweep a base curve; and 4) sew adjacent curves. They will be described in detail in this section.

3.1. Create base curve

The base curve B is defined as either a closed curve or an open curve, which lies on the surface of object Φ, $\mathrm{B} = \{e_i, e_i \in \Phi\}$, where e_i are the connected triangular edges (in order to maintain the constructed surface to be manifold, in the open curve case, e_i should be boundary edges). In our extrusion algorithm, the curve is created by the first input stroke. When users draw a stroke on the object surface, the system projects the stroke onto the surface of Φ along the view direction. In the case of an open stroke, we use the input stroke to select boundary edges to form B; and in the case of a closed stroke, we apply the partial re-triangulation algorithm to re-triangulate the object by the projected stroke (Fig. 2).

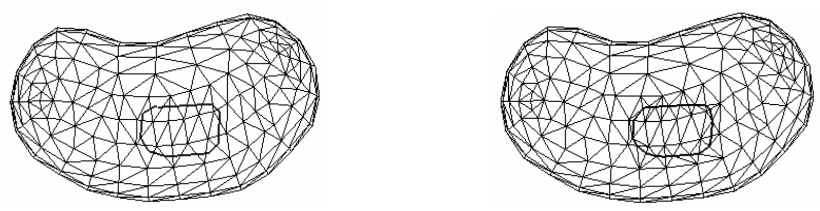
(a) user defined strokes (b) partial re-triangulation result

Fig. 2. Partial mesh re-triangulation

The whole procedure of re-triangulation consists of four steps: 1) calculate the intersections of the painted curves; 2) create new vertices and edges by painted curves; 3) triangulate triangles in $\Delta_{i,i=1,2,\ldots,n}$ one by one; 4) use the triangulation result to obtain the new object – Φ'. After projecting curves on the surface of Φ, the projected curves are represented by several 3D line segments. However, the intersections of these projected curves are not stored in our system. Thus, we need to calculate the intersections and insert them into the projected curves as nodes in the first step of re-triangulation. In the second step of re-triangulation, we use the nodes of the projected curves to create new vertices of mesh object Φ; and use the line segments of the projected curves to create new edges. Triangular edges crossed by the projected curves are divided into several edges. These new vertices and new edges should be stored in the data structure of Φ since they will be used in the third step of subdivision. The divided triangular edges are removed from Φ. The third step of re-triangulation involves subdividing the triangles in $\Delta_{i,i=1,2,\ldots,n}$ one by one; the procedure is shown in Fig. 3 (from the left column to the middle column). Because edges and vertices have been created in step two, only new triangular faces are created in this step. Here the *Constrained Delaunay Triangulation* algorithm is performed. At the end of the third step, we obtained new triangles $\Delta'_{i,i=1,2,\ldots,n}$. The fourth step is shown in Fig. 3 (from the middle column to the right column). After replacing the triangles $\Delta_{i,i=1,2,\ldots,n}$ by $\Delta'_{i,i=1,2,\ldots,m}$ in Φ, we obtain a new mesh object Φ'.

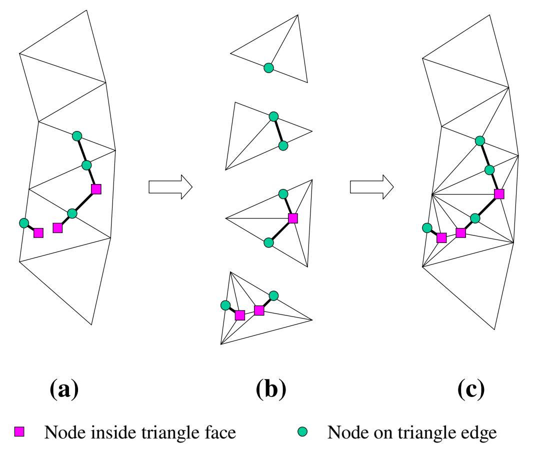

Fig. 3. Re-triangulate $\Delta_{i,\, i=1,\ldots,n}$ to obtain Φ'

After the re-triangulation, the projected curves on Φ become the edges of the triangular mesh Φ'. These triangle edges are e_i in B, they form the base curve B. The triangles inside B should be removed from Φ'.

3.2. Project extruding strokes

As shown in Fig. 4, every 2D extruding stroke Ω (there is only one stroke in the closed surface extrusion case; and there are two strokes in the open surface extrusion case) is projected onto a plane to obtain the projected 3D extruding stroke Ω_p. Therefore, the major problem of this section is how to determine the projection plane.

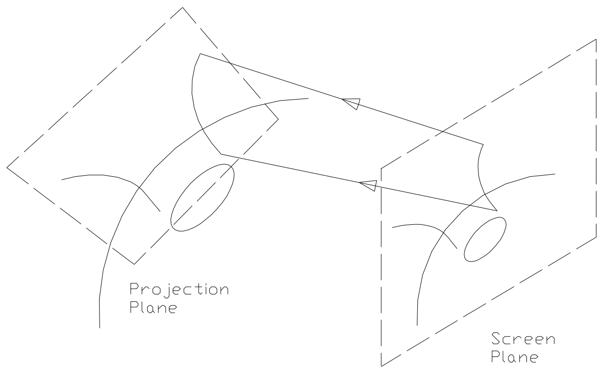

Fig. 4. Project extruding stroke

Two constraints are defined to determine the projection plane,

Constraint 1 The plane passes through two points: $P_1^*(x_1, y_1, z_1)$ and $P_2^*(x_2, y_2, z_2)$, where $P_i^* \in \mathrm{B}\ (i = 1,2)$ are the closest points to the endpoints of Ω on the screen plane.

Constraint 2 The plane lies parallel to the normal of the base curve B.

Under the above two constraints, the plane faces towards the camera as much as possible. Since we know the positions of the points $P_i^* \in \mathrm{B}\ (i = 1,\dots,n)$, we use the *Least Square* method [18] to obtain the least square plane of these points, and we use the normal of the least square plane as an approximate normal $\vec{n}_\mathrm{B} = l_\mathrm{B}\vec{i} + m_\mathrm{B}\vec{j} + n_\mathrm{B}\vec{k}$ of B. The equation of the projection plane can be written as

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ l_\mathrm{B} & m_\mathrm{B} & n_\mathrm{B} & 0 \end{vmatrix} = 0. \tag{1}$$

3.3. Sweep base curve

After the system projects the 2D extruding strokes onto the projection plane, it produces 3D extruding curves. Copies of the base curve are created along the extruding curve in such a way as to be almost perpendicular to the direction of the extrusion, and are resized to fit within the curve (Fig. 5). This is done by advancing two points (left and right) along the extruding curve starting from both ends (Fig. 6).

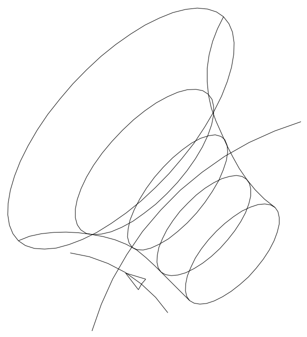

Fig. 5. Sweep base curve

In each step, the system chooses the best of the following three possibilities: advance the left point – case one, the right point – case two, or both – case three (Fig. 6). The goodness value is increased when the length of the line connecting the points is shorter. In other words, $l_1 = \|P_2P_3\|$, $l_2 = \|P_1P_4\|$, and $l_3 = \|P_3P_4\|$, where l_1, l_2, and l_3 respectively represent the line length of case one, two, and three. Thus, the possibility of the three cases – Γ_i is

$$\Gamma_i \propto \frac{1}{l_i}, \quad i = 1,2,3\,. \tag{2}$$

This process is completed when both the left and right points meet ends.

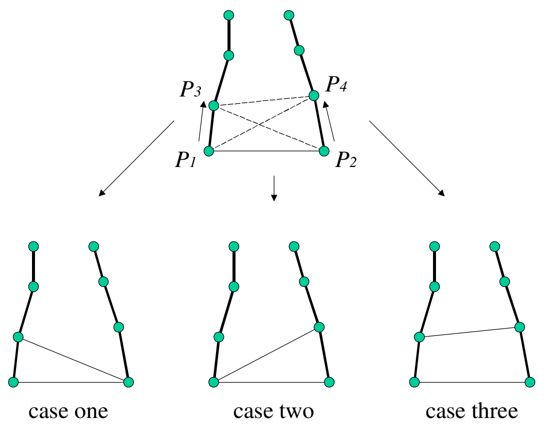

Fig. 6 From left to right: case 1, 2, and 3

After the two points are advanced, the left point and the right point are P_1' and P_2'. In case one, $P_1' = P_3$ and $P_2' = P_2$; in case two, $P_1' = P_1$ and $P_2' = P_4$; and in case three, $P_1' = P_3$ and $P_2' = P_4$. Assuming that the normal of the projection plane is $\vec{n}_p$, then the point P_1^* and the three vectors: $\vec{n}_p$, $\overrightarrow{P_1^*P_2^*}$, and $\vec{n}_p \times \overrightarrow{P_1^*P_2^*}$ build an orthogonal coordinate O^*; and point P_1' and three vectors: $\vec{n}_p$, $\overrightarrow{P_1'P_2'}$, and $\vec{n}_p \times \overrightarrow{P_1'P_2'}$ build an orthogonal coordinate O'. Briefly, the procedure of copying the base curve B is to translate the vertices on B from coordinate O^* to coordinate O', and scale them by the ratio $\frac{\left\|\overrightarrow{P_1'P_2'}\right\|}{\left\|\overrightarrow{P_1^*P_2^*}\right\|}$. To construct a closed surface, the base curve B should be a ring, and the last copied ring is degenerated to a single vertex.

3.4. Sew adjacent curves

In this step of mesh surface extrusion, we create new polygons by sewing the neighboring copies of the base curve together [3, 4] (Fig. 7). While sewing the adjacent curves, we should preserve the orientation of the created triangular faces. Firstly, we define that 1) if an edge is stored in the anti-clockwise direction in a triangular face, it is called a positive edge; otherwise, it is called a negative edge – for example, in Fig. 8, edge e is positive in face f_1 and negative in face f_2; 2) an edge with only one triangle adjacent is called a boundary edge. The orientation of the created faces is maintained by the following rules.

Rule 1 Only triangles with boundary edges can be constructed;

Rule 2 Make each triangular edge positive in one adjacent face and negative in another adjacent face when determining the orientation of a triangular face.

Rule 1 gives the order of constructing triangles: if in two adjacent curves B_1 and B_2, B_1 is the previous one of B_2 while sweeping the base curve, we firstly construct the triangles with edges in B_1 and secondly construct the triangles with edges in B_2; and we begin the construction from the triangles with edges in the base curve. When constructing a triangular face, we use rule 2 to determine its orientation. These two rules preserve the orientation of triangles in the extrusion result.

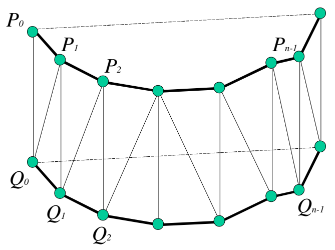

Fig. 7. Sew neighboring rings

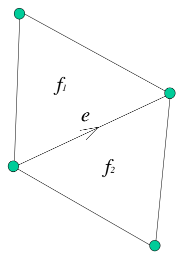

Fig. 8. Direction of an edge in adjacent faces

After sewing the adjacent curves, a new 3D polygonal object is created (e.g., the model shown in Fig. 9a). Analyzing the mesh representation of the 3D polygonal object, we find that there are many short edges and small triangles in the extruded mesh surface. Here, we implement a mesh optimization algorithm [1] to remove these short edges and small triangles; the proposed mesh optimization result is shown in Fig. 9b.

(a) before optimization **(b) after optimization**

Fig. 9. Mesh optimization

4. Examples and Applications

We are developing a sketch based 3D freeform modeling system now. The sketch based freeform extrusion approach presented in this paper has been implemented as a part of our prototype system. The mesh extrusion can be finished in real-time on a standard PC. Users can create a wide variety of shapes using our mesh extrusion tool (e.g., Fig. 10a – long shape; Fig. 10b – thin shape; Fig. 10c – sharp shape); and they can use the same tool to dig a cavity on the surface (Fig. 10d).

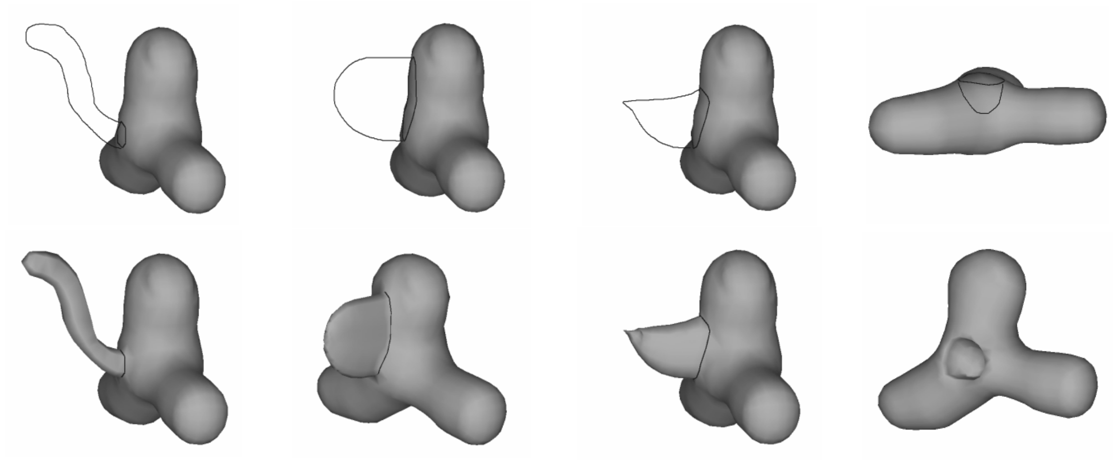
(a) long **(b) thin** **(c) sharp** **(d) cavity**

Fig. 10. Examples of extrusion

Fig. 11 shows an application for toy design, where the mouth, the tail and the wings of a duck are created from the initial object (shown in Fig. 11a) by using our mesh extrusion tool. Other sketch based 3D freeform modeling tools of our prototype system construct the initial object. Note that the model looks quite different from 3D models created by other modeling systems – it reflects the hand-drawn nature of the shape.

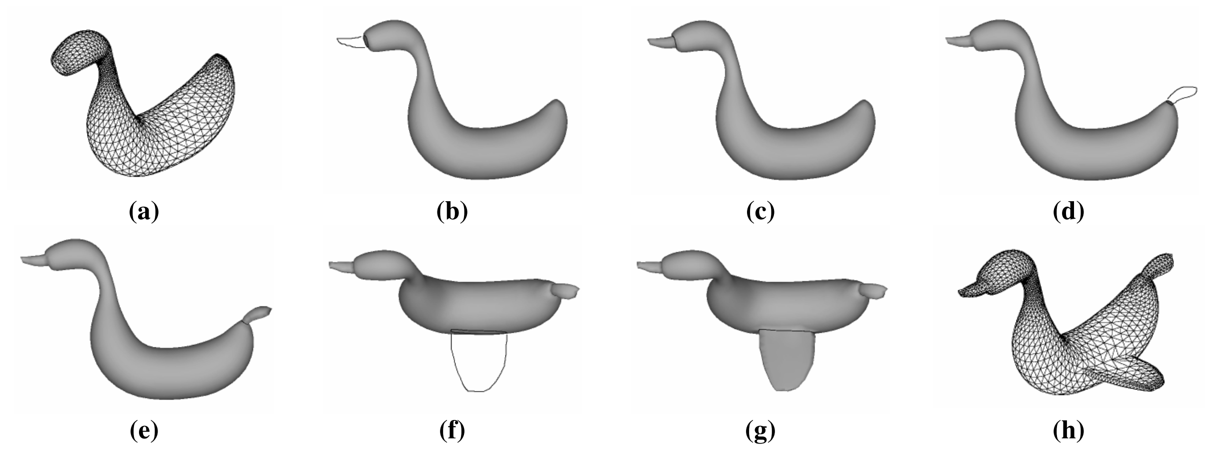

Fig. 11. Extrusion application one: toy design

Fig. 12 shows another application that is for fashion design: in Fig. 12a, two strokes are input to extrude the lower hem of the given waistcoat (which is constructed by other tools in our prototype system) to construct a dress (as shown in Fig. 12b); in Fig. 12c, the whole armhole is extruded to form a band sleeve (as shown in Fig. 12d); in Fig. 12e, a part of the armhole is extruded to construct a cape sleeve; and in Fig. 12g, our extrusion tool is used to change the collar of the dress from the scoop style to the ring style (as shown in Fig. 12h). The shading representation of the dress is shown in Fig. 12i; after applying the draping simulation on the dress by the 3D Garment v2.0 system [19], the result is shown in Fig. 12j, 12k, and 12l.

From the above examples and applications, it is easy to find that our extrusion approach is an intuitive, efficient geometric modeling tool for freeform polygonal object modeling. The number of vertices and faces of the toy design application and the fashion design application is shown in Table 1. Both the number of vertices and the number of faces are increased considerably after extrusion. Our approach can complete the mesh extrusion in real-time on a standard PC, but may cause a short pause (a few seconds) when the model becomes complex – the number of vertices and faces increases.

Table 1 Increase of the mesh size

Example	**Before extrusion**		**Final result**	
	Vertices number	**Triangles number**	**Vertices number**	**Triangles number**
Toy design (Fig. 11)	1,616	3,228	2,093	4,180
Fashion design (Fig. 12)	399	712	1,288	2,476

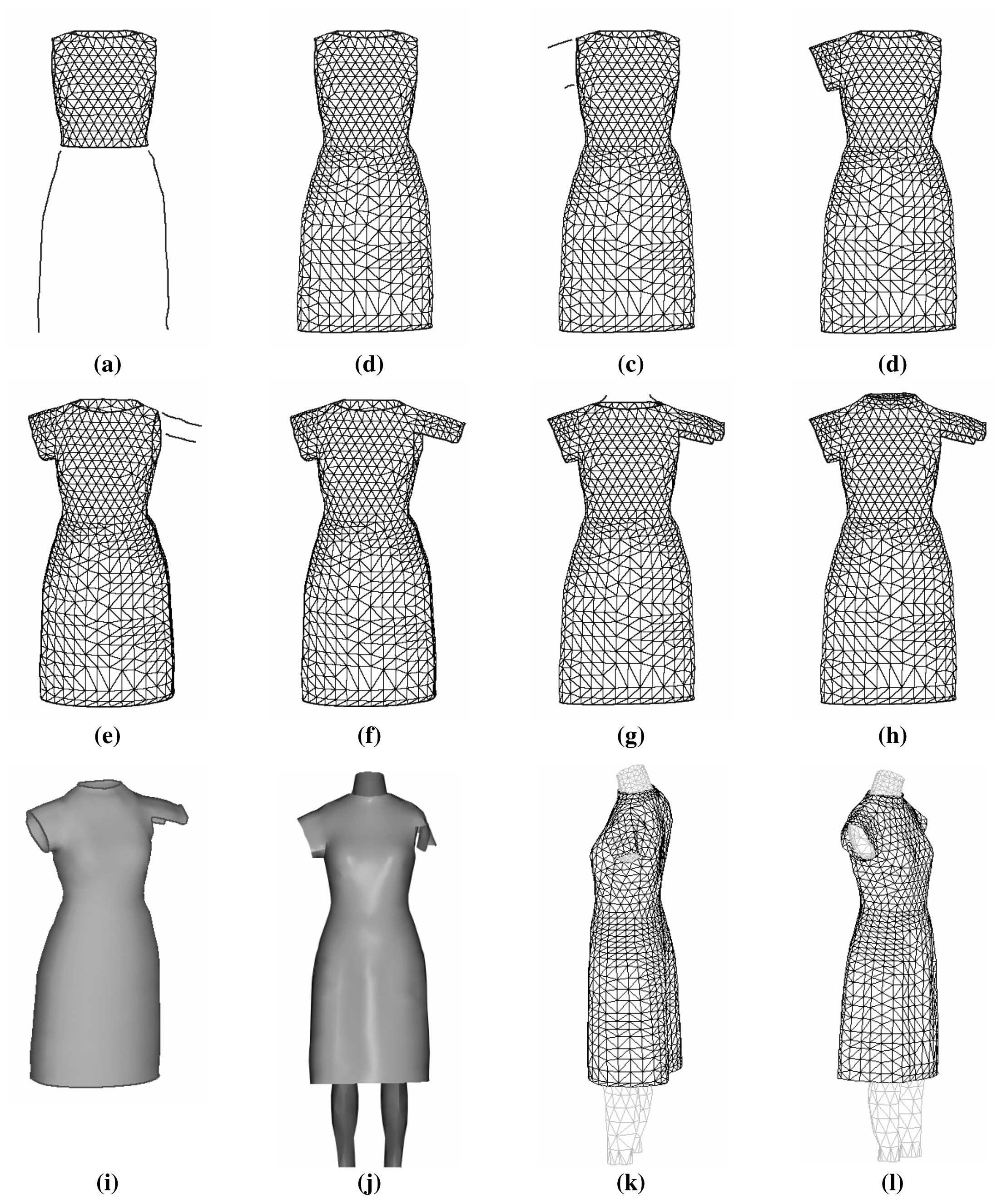

Fig. 12. Extrusion application two: fashion design

5. Conclusion and Discussion

In this paper, a sketch based mesh extrusion method, which is useful for intuitive, efficient geometric modeling of freeform polygonal objects, is introduced. Our extrusion approach is related to the extrusion method in the Teddy system. With our method, users can extrude either a closed or open surface from the surface or the boundary of a given polygonal mesh by sketched input. However, the Teddy system can only extrude a closed surface from the surface of an initial mesh. Thus, compared with earlier method, the spherical topology limitation is overcome. Our mesh extrusion algorithm consists of four steps: 1) create base curve; 2) project extruding strokes; 3) sweep base curve; and 4) sew adjacent curves. At the end of this paper, examples of our mesh extrusion implementation are shown to demonstrate its functionality. Our freeform extrusion has been implemented as a part of our sketch based modeling system, and the mesh extrusion can be completed in real-time on a standard PC.

Our current algorithm is robust and efficient enough for experimental use. However, they can fail or generate unintuitive results when the user draws unexpected strokes (i.e., the implementation of extrusion does not support holes that completely extend to the other side of the object). The problem of getting a 3D shape of a surface from its contour is a problem with insufficient constraints; so our algorithm uses some assumption as additional constraints when constructing an extruded surface. If these assumptions do not fit the requirement of users, our algorithm gives incorrect results. Also, since the specification of the base curve is based on the projection along the view direction like ray tracing, we cannot define based curves on the invisible surface from the screen. Our present implementation cannot avoid the intersection between the extruded surface and the existing surface. In particular, we are planning to enhance the extrusion algorithm by integrating collision detection and self-collision detection algorithms. Another research direction should focus on how to extrude a non-manifold surface (e.g., a surface like Möbius Strip, which is usually used when designing a wedding dress).

Acknowledgements

The authors would like to acknowledge the comments given by Mr. Danny Cheung, and the support of the CAD/CAM Facility of the Hong Kong University of Science and Technology.

References

[1] Wang CCL, Yuen MMF. A generic algorithm of mesh optimisation, International Journal of Advanced Manufacturing Technology, vol. 18, no. 10, 2001, pp. 739-744. Publisher: Springer-Verlag, UK.

[2] Igarashi T, Tanaka H, Matsuoka S. Teddy: A Sketching Interface for 3D Freeform Design, SIGGRAPH 1999 Conference Proceedings, pp.409-416, 1999. ACM., New York, NY, USA.

[3] Meyers D, Skinner S, Sloan K. Surface from Contours, ACM Transaction on Graphics, vol. 11, no. 3, 1992, pp. 228-258.

[4] Keppel E. Approximating complex surfaces by triangulation of contour lines, IBM Journal Res. Develop, Jan. 1975, pp. 2-10.

[5] Hoppe H, DeRose T, Duchamp T, Halstead M, Jin H, McDonald J, Schweitzer J, Stuetzle W. Piecewise smooth surface reconstruction, SIGGRAPH 1994 Conference Proceedings, pp.295-302, 1994. ACM., New York, NY, USA.

[6] Zeleznik RC, Herndon KP, Hughes JF. SKETCH: An interface for sketching 3D scenes, SIGGRAPH 1996 Conference Proceedings, pp. 163-170, 1996. ACM., New York, NY, USA.

[7] Bloomenthal J, Wyvill B. Interactive techniques for implicit modeling, 1990 Symposium on Interactive 3D Graphics, pp. 109-116, 1990.

[8] Bloomenthal J. Introduction to implicit surfaces, San Francisco, C.A.: Morgan Kaufmann Publishers, Inc., 1997.

[9] Markosian L, Cohen JM, Crulli T, Hughes J. Skin: a constructive approach to modeling free-form shapes, SIGGRAPH 1999 Conference Proceedings, pp. 393-400, 1999. ACM., New York, NY, USA.

[10] Suzuki H, Sakurai Y, Kanai T, Kimura F. Interactive mesh dragging with an adaptive remeshing technique, Visual Computer, vol.16, no.3-4, 2000, pp.159-76. Publisher: Springer-Verlag, Germany.

[11] Zorin D, Schroder P, Sweldens W. Interactive multiresolution mesh editing, SIGGRAPH 1997 Conference Proceedings, pp. 259-68, 1997. ACM., New York, NY, USA.

[12] Khodakovsky A, Schroder P. Fine level feature editing for subdivision surfaces, Proceedings Fifth Symposium on Solid Modeling and Applications, Jun. 1999, pp.203-11. N.Y.: ACM Press, USA.

[13] Schroeder WJ, Zarge JA, Lorensen WE. Decimation of triangle meshes, Computer Graphics, vol. 26, n2, 1992, pp.65-70, USA.

[14] Garland M, Heckbert PS. Surface simplification using quadri error metrics, SIGGRAPH 1997 Conference Proceedings, pp. 209-16, 1997. ACM., New York, NY, USA.

[15] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Mesh optimization, SIGGRAPH 1993 Conference Proceedings, pp.19-26, 1993. ACM., New York, NY, USA.

[16] Frey PJ, Borouchaki H. Geometric surface mesh optimization, Computing & Visualization in Science, vol.1, no.3, Nov. 1998, pp.113-121, 1998. Publisher: Springer-Verlag, Germany.

[17] Kobbelt LP, Bareuther T, Seidel HP. Multiresolution shape deformation for meshes with dynamic vertex connectivity, Computer Graphics Forum, vol. 19, no. 3, Eurographics' 2000, pp. 249-259, 2000.

[18] Dahlquist G, Bjorck A. Numerical Method (translated by Anderson N), Englewood Cliffs, N.J.: Prentice-Hall, 1974.

[19] 3D Garment v2.0, 2000, Hong Kong University of Science and Technology, Hong Kong.