

An Interactive Sketch-based Modeling System using a Topology Library and Subdivision Methods

Weizhong Liu[†], Kunio Kondo[†] and Jun Mitani[‡]

[†] Department of Information and Computer Sciences, Saitama University.

Email: {wzliu, kondo}@ke.ics.saitama-u.ac.jp

[‡] Graduate School of Systems and Information Engineering,
University of Tsukuba. Email: mitani@cs.tsukuba.ac.jp

Abstract

We present a sketch-based modeling system for interactive modeling of a variety of 3D objects. The idea is to construct and modify 3D objects, step by step, using a 2D template topology library. The library, called T-LIB, consists of 2D Edge Graphs and the corresponding algorithms to construct and modify 3D objects. The local part of 3D object around the sketch lines is analyzed as a 2D Edge Graph. Once the Edge Graph can be matched to a template of T-LIB, the 3D object can be generated in a basic shape reconstruction procedure, or modified in a 3D shape modification procedure. After obtaining the solid model, we use a fillet operation and a subdivision method to generate smooth surface objects.

Keywords: Sketch, Geometric Modeling/Reconstruction, Template, Edge Graph, Subdivision method/Surfaces

ACM CSS: I.3.5 [Computer Graphics]: Constructive solid geometry (CSG) and Geometric algorithms, languages, and systems. I.3.6 [Computer Graphics]: Interaction techniques

1. Introduction

Currently, much of CAD software and similar modeling systems can display a 3D object model created by designers, but these advanced 3D systems demand precise data of the 3D object, especially in the steps of inputting data and making freeform surfaces. They are difficult and tedious for designers to use. On the other hand, traditional sketching is usually more efficient in the early stages of concept design, compared with traditional CAD systems. Recently sketch systems have been introduced to bridge the gap between concept design and computer-based modeling programs, combining some of the features of pencil-and-paper sketch with features of CAD systems, to provide a lightweight interface to “approximate” 3D polyhedral modeling.

This paper extends these ideas to propose a sketch-based modeling system for interactive modeling of a variety of 3D solid objects and surface objects.

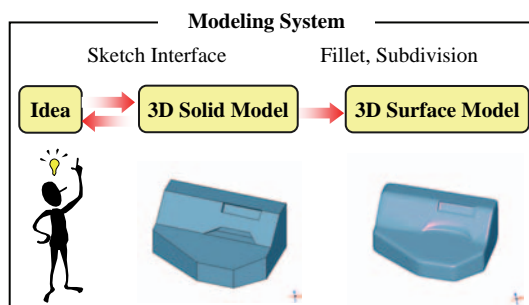


Figure 1: Sketch-based modeling system.

This paper pursues the concept of creating a complex 3D object using a topology library. The library, called T-LIB, consists of 2D Edge Graphs and the corresponding algorithms to modify 3D objects. The local region of 3D object around the sketch lines is analyzed as a 2D Edge Graph. Once the Edge Graph can be matched to a template from T-LIB, the 3D object can be generated in a basic shape reconstruction procedure, or modified in a 3D shape

modification procedure. The modeling system is illustrated in Fig. 1, while the sketch interface is shown as Fig. 2.

In our proposed system, we aim to roughly construct a 3D object with several steps for fast evaluation of the ideas of a designer. The system is very useful in applications for modeling “prismatic” 3D objects. The application areas would be in industrial design, concept design by designers, education, etc.

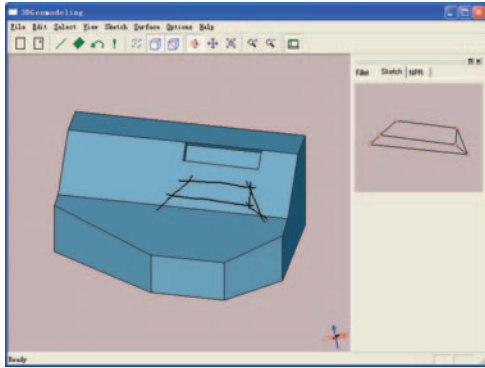


Figure 2: Sketch interface.

2. Related Works

Sketch-based systems are a relatively new area in modeling. The main goal of sketch-based systems is to allow the creation of 3D models by using strokes extracted from user input and/or existing drawing scans.

We can distinguish five approaches, as shown in Fig. 3, depending on the methods used to recognize and recover the 3D object from a 2D sketch. Each approach can be classified as two types: polygon-based and freeform-based. The first approach uses one projection to recover the 3D object from 2D edge graph. The second approach constructs 3D object by editing a basic object: box. The third approach uses outlines (silhouette curve) to create the 3D object. The fourth approach uses primitive objects to construct a 3D scene. Finally, the fifth approach constructs the 3D objects from several cross sections. These approaches and the related work will be introduced in the following sub-sections.

2.1 One Projection approach

A one-projection approach uses the theory of perspective projections to calculate the coordinates of the vertices of a 3D object. The related work can be found in [Hosaka 1983], [Fukui 1988], [Kondo 1988], [Mitani 2000], [Shesh 2004], etc.

Mitani[2000] proposed a 3D sketch system with a simple template which uses one projection and applies mirror

symmetry to reconstruct a surface model. But only a single hexahedral model can be constructed.

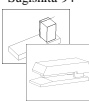






	One Projection	Box	Outline	Primitive	Cross Section
Polygon	Hosaka'83 Fukui'88 Kondo'88 Lamb'90	Sugishita'94 	Varley'2000 Naya'2002	Zelevnik'96 Shesh'2004 	Furushima'90 Akeo'94 
Freeform	Mitani'2000 	Xu'95 	Igarashi'99  Cherlin'2005		Matsuda'2000  Kuragano'2002

Figure 3: Category of sketch system.

2.2 Box approach

The box approach is based on a basic object: cube (box). The complex object can be generated by modifying and combing the basic objects. This work has been presented by [Sugishita 1994], [Xu 1995], etc.

Sugishita[1994] proposed a sketch-based modeling system. The system creates a basic cube-like shape. The basic shape is operated on by either adding parts or cutting off pieces from the basic shape. A cutting plane is decided from three points which are the intersections of additional cutting lines with a basic shape. But the modeling process is tedious for the designer.

2.3 Outline-based approach

The outline-based method is widely used in sketch-based modeling. Many well-known systems are based on this method, such as [Igarashi 1999], [Varley 2000], [Naya 2002], [Cherlin 2005], etc.

Varley[2000] proposed RIBALD, which can convert a sketch into a B-Reps solid model. This method assumes that the output mesh is geometrically similar to a pre-defined template. However, several assumptions must be made about sketches in order for RIBALD to perform this conversion, and thus it cannot directly accept a sketch input by the designer.

Teddy[Igarashi1999] is a sketch based system that allows the user to easily create free-form 3D models. The system allows the creation of a surface by inflating regions defined by closed strokes. Strokes are inflated, using a chordal axis transform, so that portions of the mesh are elevated, based on their distance from the stroke's chordal axis. Teddy also allows users to create extrusions, pockets and cuts to enable the editing of the models in quite flexible ways. However, it is difficult to model a prismatic object.

2.4 Primitive approach

The primitive-based method uses several primitive objects to construct the whole 3D scene. Related work is by [Zelevnik 1996], [Shesh 2004], etc.

Zelevnik[1996] introduced a gesture-based interface for the rapid modeling of CSG-like models, consisting of simple primitives. The system aims at architectural forms, in which the geometric model is entered by a sequence of gestures according to a set of conventions, taking regard to the order in which points and lines are entered as well as their spatial relations. For example a primitive of a block type is defined by three segments, starting from the same point. Positive volumes are built in the same direction as the outer normal to an adjacent surface, whereas negative volumes are drawn opposing the outer normal. However modeling process is not intuitive.

2.5 Cross Section approach

A Cross Section is a planar curve on the surface of a 3D object. The 3D object can be reconstructed from several cross section curves. Related work has been reported by [Akeo 1994], [Matsuda 2000], [Kuragano 2002], etc.

Akeo[1994] proposed a method for construction of a 3D model by inputting a 2D sketch in a computer using an image scanner. A section view of the model is added by the designer in order to supply additional information to reconstruct the 3D model. The designer can use standard sketching tools such as a pencil and paper, although this interface is not interactive.

Matsuda[2000] proposed freehand modeling, based on cross section method. The system creates a control polygon for each freehand cross section curve, and then generates polyhedral networks by the product cluster from a cylinder that sweeps the section from each shaft direction, and finally uses a subdivision technique to generate a 3D surface model. Three cross section curves are used to construct a 3D shape, and these curves are assumed to be orthogonal. Only a simple rough object can be generated, and the system does not support editing.

Our system is based on the box method. With T-LIB, the 3D geometric components can be constructed or modified directly with sketch lines. Compared with traditional CAD systems, the features of our system can be described as follows:

- (1) It does not require precise data input.
- (2) The modeling process is very simple.
- (3) It is intuitive.
- (4) The modeling process is similar to the designing process used by the designer.
- (5) The system is extendable since T-LIB is not limited.

3. System Overview

3.1 Motivation

Our motivation to realize such a system is illustrated in the following figures and process [Fig. 4]. The whole system consists of three main sections. In the first section polyhedrons are generated from freeform sketch lines with a sketch interface. This is the process for solid modeling. The second section generates complex polyhedra by a process called fillet operation. The last section constructs free form surfaces by subdivision methods.

It is mainly the first section that is discussed in this paper. The fillet operation can be referred to in Liu[2003], while the subdivision can be referred to in Liu[2004].

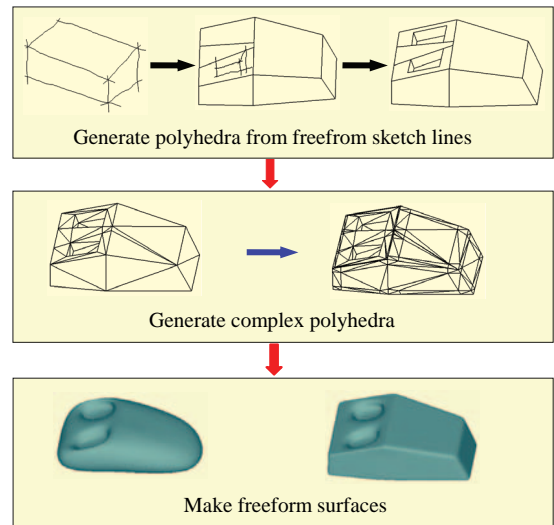


Figure 4: Motivation.

3.2 System Procedure

The system procedure follows the route shown in Fig. 5. It consists of 5 steps:

- (1) Edge Graph analysis.
- (2) T-LIB matching and new template definition.
- (3) 3D shape reconstruction.
- (4) 3D shape modification.
- (5) Fillet operation, subdivision.

Step (1, 2) will be discussed in section 4. Step (3, 4) will be discussed in section 5. Step (5) is referenced in the related research work, Liu[2003] and Liu[2004].

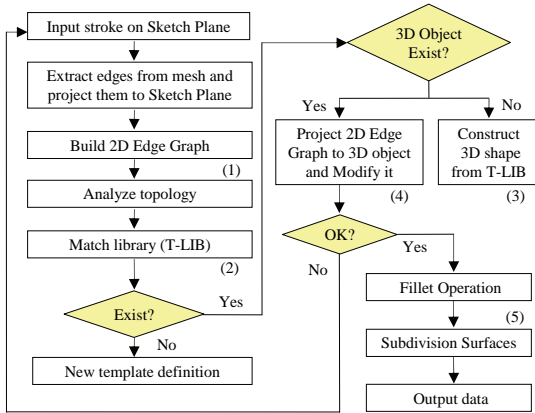


Figure 5: System procedure.

3.3 User Interface

Line strokes are drawn on a plane which is called Sketch Plane. This can be regarded as a plane between 3D object space and the screen of the user interface. Referring to Fig. 6, there is a cube in 3D object space. The designer draws strokes on the 3D cube in order to modify it. However, the strokes are actually drawn in the Sketch Plane.

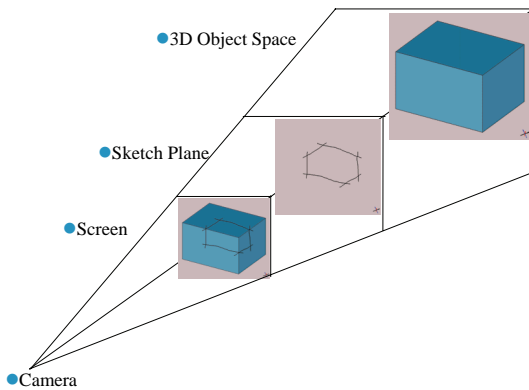


Figure 6: User Interface.

4. Analysis the input of sketch lines

4.1 Edge Graph

The Edge Graph is a 2D topology graph. It is calculated from the input strokes and the projected edges of a 3D mesh which intersects these strokes. Here, we use a proper threshold to judge whether the strokes meet at one vertex or not.

4.1.1 Edge Graph analysis case 1

Referring to Fig. 7, case 1 is where no edge of 3D mesh intersects the input strokes. In this case, we will build a 2D Edge Graph directly.

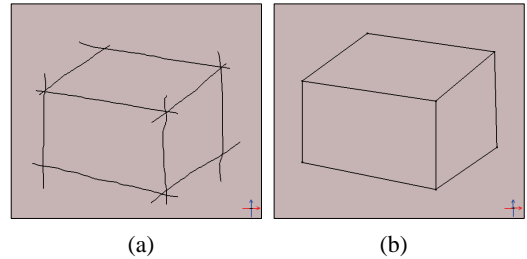


Figure 7: Edge Graph analysis case 1. (a) Input stroke. (b) Build 2D Edge Graph from (a).

4.1.2 Edge Graph analysis case 2

Referring to Fig. 8, case 2 is where some edges of the 3D mesh intersect the strokes. In this case, we implement a process called T-EXTRACT to obtain the intersecting edges from the 3D mesh. These are then projected onto the Sketch Plane. From the input strokes, together with the projected edges, we can build a 2D Edge Graph.

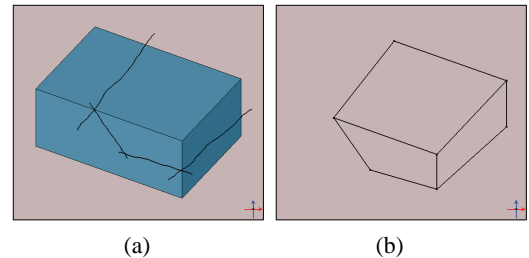


Figure 8: Edge Graph analysis case 2. (a) Input stroke. (b) Build 2D Edge Graph from (a).

4.1.3 About noise

Noise is caused by illegal stroke input and illegal extracted edges of 3D mesh in section 4.1.2. Thus, in the Edge Graph analysis process, we should implement a process to remove the noise strokes and noise extracted edges of 3D mesh.

4.2 T-LIB

T-LIB is a template topology library. It consists of 2D Edge Graphs and the corresponding algorithms to create and modify 3D objects. For the present, in our system, we

only consider the following simple templates: cube, triangular prism and triangular pyramid.

Of course, the templates are not limited to only these three. Based on the requirements of the designer, new template definition is feasible.

4.2.1 Explicit templates

Here, we use the term “explicit” which means all edges of Edge Graph are visible by the camera. The explicit templates are illustrated in Fig. 9. They are used both in 3D shape reconstruction and modification processes.

T-LIB	2D Edge Graph		3D Shape
Cube			
Triangular Prism			
Triangular Pyramid			

Figure 9: T-LIB1 (Explicit templates).

4.2.2 Implicit templates

Here, we use the term “implicit” which means some edges of the Edge Graph are invisible to the camera. The implicit templates are illustrated in Fig. 10. They are used only in the modification process.

T-LIB	2D Edge Graph	3D Shape
Cube		
Triangular Prism		

Figure 10: T-LIB2 (Implicit templates).

5. Algorithm and modeling process

5.1 3D shape reconstruction

The algorithm of cube shape reconstruction is discussed in this section. The input strokes are shown in Fig. 11(a), while the extracted 2D Edge Graph is shown in Fig. 11(b). Fig. 11(c) shows the 3D Edge Graph. The 3D cube (Fig. 11(d)) is calculated by evaluating a camera normal and using parallel projection.

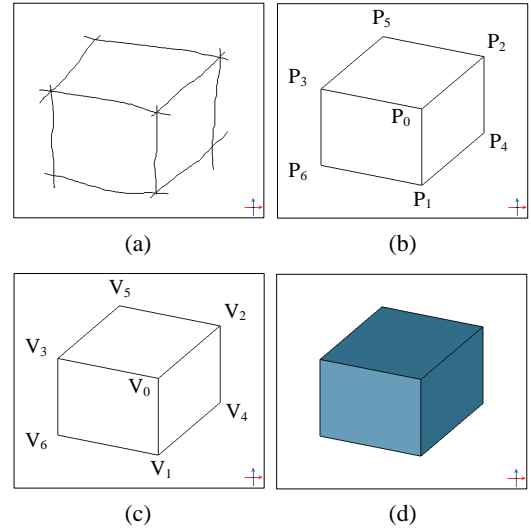


Figure 11: 3D cube shape reconstruction.

Let vector $n = \{n_x, n_y, n_z\}$ be the camera normal, $V_i = \{v_i^x, v_i^y, v_i^z\}$ be the vertex of the 3D cube, while $P_i = \{p_i^x, p_i^y, p_i^z\}$ is the vertex of 2D cube Edge Graph. Obviously, $p_i^z = 0$. P_0 is chosen as the center of the world frame. $P_0 = (0,0,0)$.

Referring to Fig. 11(b) and Fig. 11(c), with the principle of parallel projection, we obtain the following equations:

$$\frac{v_1^x - p_1^x}{n_x} = \frac{v_1^y - p_1^y}{n_y} = \frac{v_1^z}{n_z} \quad (1)$$

$$\frac{v_2^x - p_2^x}{n_x} = \frac{v_2^y - p_2^y}{n_y} = \frac{v_2^z}{n_z} \quad (2)$$

$$\frac{v_3^x - p_3^x}{n_x} = \frac{v_3^y - p_3^y}{n_y} = \frac{v_3^z}{n_z} \quad (3)$$

Referring to Fig. 11(c), letting $V_0 = (0,0,0)$, then vector $V_i V_0$ equals V_i ($i=1,2,3$). Since vector $V_1 V_0$, $V_2 V_0$ and $V_3 V_0$ are perpendicular to each other, we can obtain the following equations:

$$v_1^x v_2^x + v_1^y v_2^y + v_1^z v_2^z = 0 \quad (4)$$

$$v_2^x v_3^x + v_2^y v_3^y + v_2^z v_3^z = 0 \quad (5)$$

$$v_3^x v_1^x + v_3^y v_1^y + v_3^z v_1^z = 0 \quad (6)$$

Letting $a=n_x/n_z, b=n_y/n_z$, from formula (1-3), we can obtain the following equations:

$$v_1^x = av_1^z + p_1^x, v_1^y = bv_1^z + p_1^y \quad (7)$$

$$v_2^x = av_2^z + p_2^x, v_2^y = bv_2^z + p_2^y \quad (8)$$

$$v_3^x = av_3^z + p_3^x, v_3^y = bv_3^z + p_3^y \quad (9)$$

Letting $c=(ap_1^x+bp_1^y)/(a^2+b^2+1), d=(ap_2^x+bp_2^y)/(a^2+b^2+1), e=(ap_3^x+bp_3^y)/(a^2+b^2+1), f=(p_1^x p_2^x + p_1^y p_2^y)/(a^2+b^2+1), g=(p_2^x p_3^x + p_2^y p_3^y)/(a^2+b^2+1), h=(p_3^x p_1^x + p_3^y p_1^y)/(a^2+b^2+1)$, from equation (4-9), we obtain the following equations:

$$v_1^z v_2^z + d v_1^z + c v_2^z + f = 0 \quad (10)$$

$$v_2^z v_3^z + e v_2^z + d v_3^z + g = 0 \quad (11)$$

$$v_3^z v_1^z + c v_3^z + e v_1^z + h = 0 \quad (12)$$

From equation (10-12), we can obtain equation (13).

$$(g-de)(v_1^z)^2 + 2(cg-cde)v_1^z + (fh+gc^2-cdh-cef)=0 \quad (13)$$

Then vertex V_1 can be calculated, vertexes V_2, V_3 can be derived from equation (10-12). Other vertices can be calculated with parallel information. In this manner, the basic 3D shape can be generated.

For the triangular prism and triangular pyramid templates in T-LIB1, the algorithms of 3D shape reconstruction are also given by using the principle of parallel projection and some assumption.

5.2 3D shape modification

5.2.1 Project 2D Edge Graph to 3D object

In the 3D modification process, after calculating the 2D Edge Graph in the Sketch Plane, we project the 2D Edge Graph on to 3D object and obtain a 3D Edge Graph. The hidden face can be calculated by referring to this 3D Edge Graph.

Referring to Fig. 12, we input strokes on the 3D shape (Fig. 12(a)) and build a 2D Edge Graph (Fig. 12(b)). Then we project the 2D Edge Graph on to 3D shape and adjust the edges (Fig. 12(c)). The modified shape is shown in Fig. 12(d).

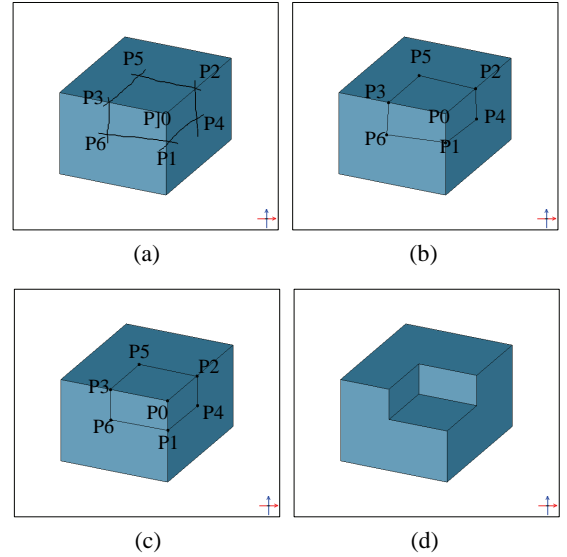


Figure 12: Modification process.

5.2.2 Modify the 3D shape

After projecting the 2D Edge Graph on to 3D object, the modification procedure functions as follows:

For each face of 3D object from T-LIB, if it is in a face f_i of the 3D shape, we modify f_i by deleting this part from f_i , while if it is not in a face f_i , we create a same new face and add it to the 3D object.

Here the angle between the stroke and edge of 3D shape is used as a threshold to adjust the Edge Graph for obtaining the parallel information.

5.2.3 About cracking problem

The modification process can produce a cracking problem if the hidden face intersects with another face of the 3D shape.

Based on the different requirements, there are two solutions to solving the cracking problem. One is automatic adjustment of the parameters, the other is the calculation of the intersecting lines and then modification of the intersecting faces.

6. Fillet operation and subdivision surfaces

6.1 Subdivision surfaces

The subdivision surface has become one of the basic tools in CAGD for modeling complex surfaces, since the polyhedral subdivision process provides a simple and efficient way to generate surfaces over polyhedral networks. We

propose an adaptive scheme for the subdivision surface, based on triangular meshes [Liu 2004].

6.2 Fillet operation

Surfaces can be generated easily by subdividing polyhedral networks recursively, but over a simple or rough polyhedral network, it is difficult to achieve the surfaces which designers wish to model. To solve this problem, we must define a more complex or precise polyhedral network.

A solution called fillet operation has been proposed by Liu[2003]. The motivation of fillet operation is illustrated in Fig. 13.

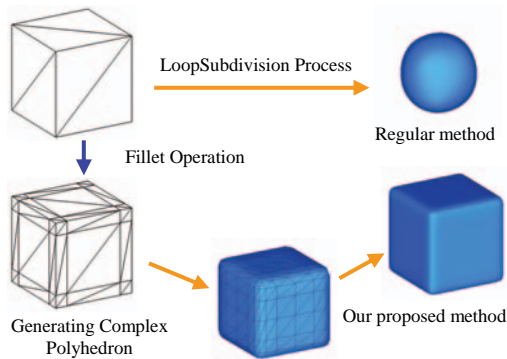


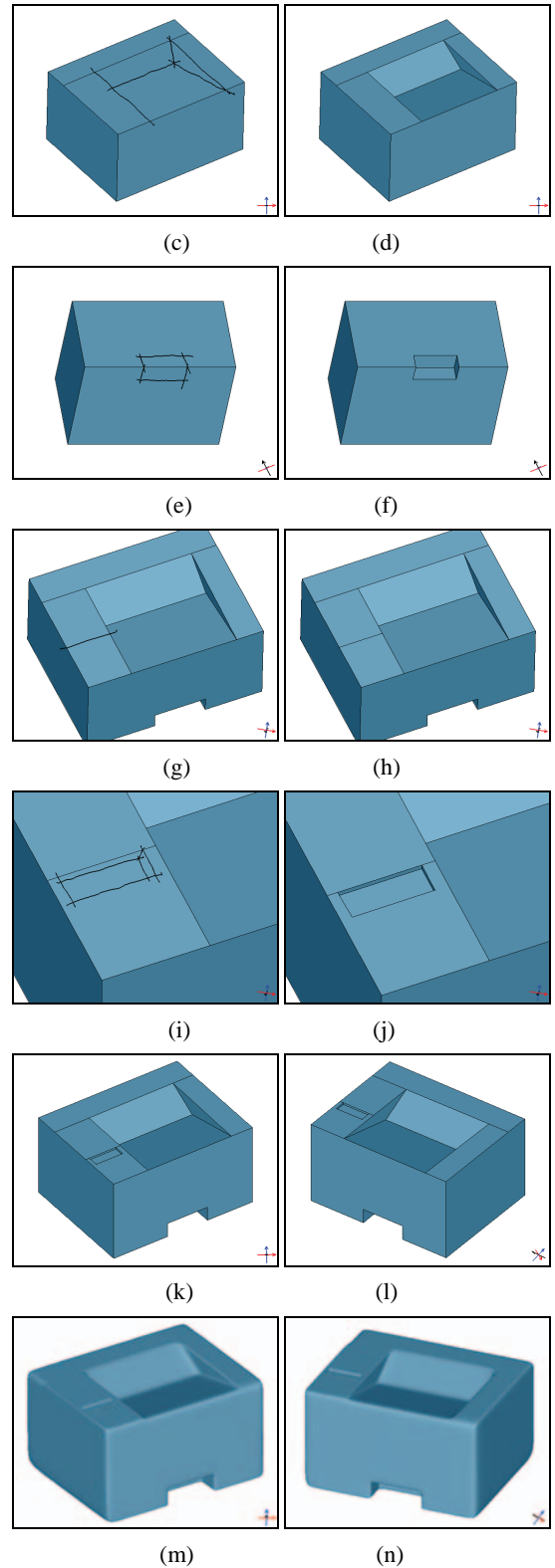
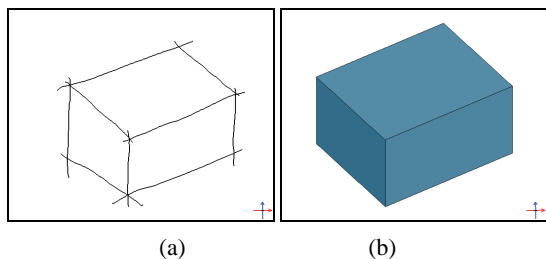
Figure 13: Fillet operation.

7. Examples

In this section, we show several models generated by our system. Each model is illustrated as a series of figures from the beginning of stroke input to completion of the final shape.

7.1 Modeling a printer-like shape

Fig. 14 is an example we made by using our system for modeling a printer-like shape. The procedures are described as follows: (a) Input strokes. (b) Create basic 3D cube. (c-l) Input strokes and modify repeatedly. (m-n) Implement fillet operation and subdivision. (o-p) Map strokes on the 3D object.



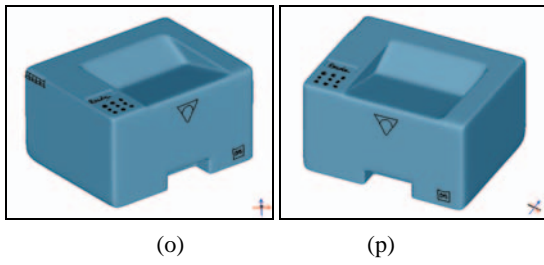


Figure 14: Modeling a printer-like shape.

7.2 Modeling a clock-like shape

Fig. 15 is an example of modeling a clock-like shape. The procedures are described as follows: (a) Input strokes. (b) Create basic 3D cube. (c-l) Input strokes and modify repeatedly. (m-n) Implement fillet operation and subdivision. (o-p) Map strokes on the 3D object.

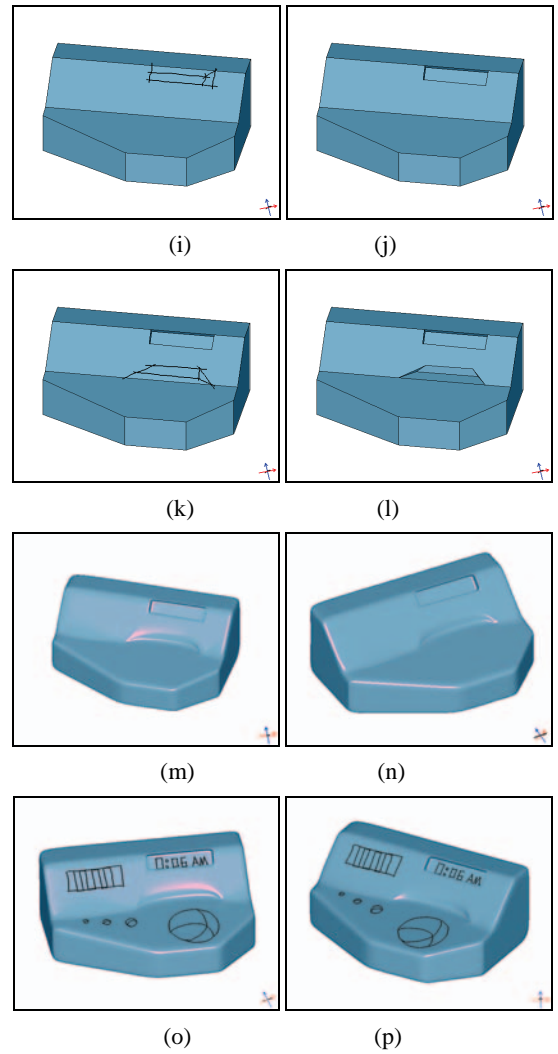
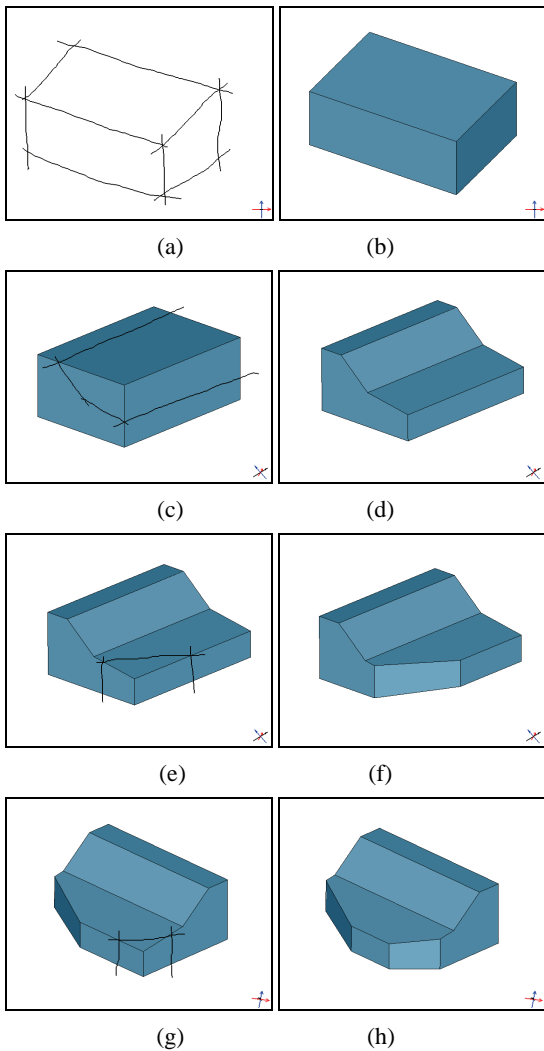
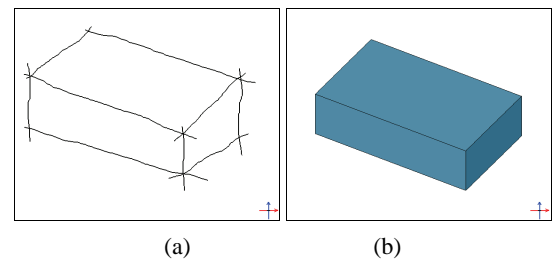


Figure 15: Modeling a clock-like shape.

7.3 Modeling a mouse-like shape

Fig. 16 is an example of modeling a mouse-like shape. The procedures are described as follows: (a) Input strokes. (b) Create basic 3D cube. (c-j) Input strokes and modify repeatedly. (k-l) Implement fillet operation and subdivision.



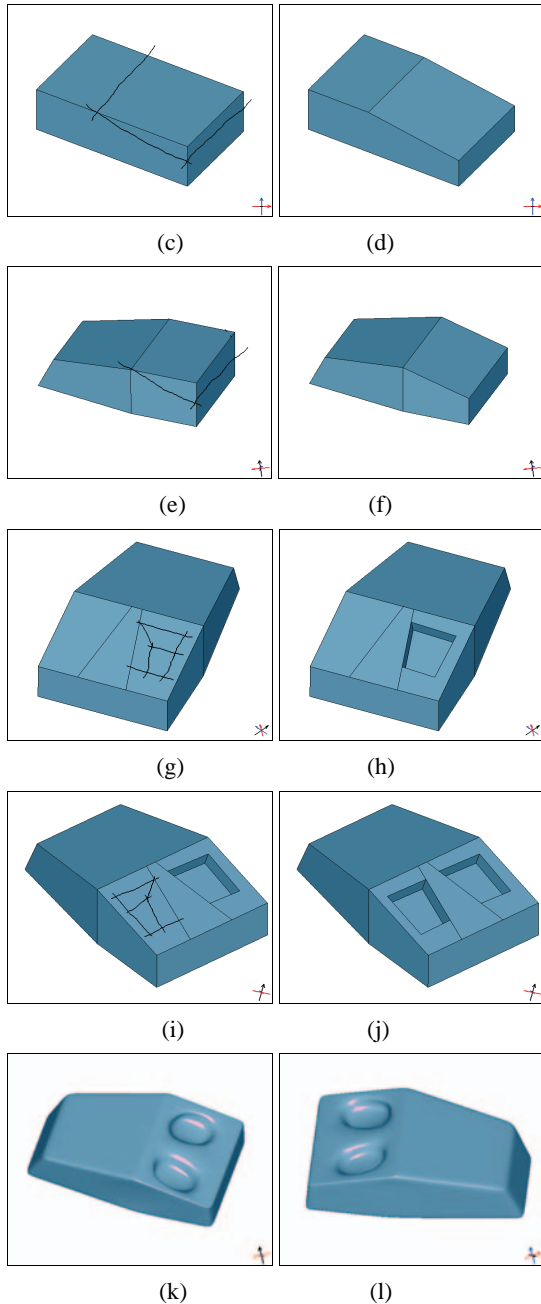


Figure 16: Modeling a mouse-like shape.

7.4 Evaluation

From the examples above, it can be seen the modeling is quite simple. The user can make such 3D objects within several minutes. Only few works are required of the user to learn. The works are described as follows.

- (1) Knowledge of modeling. (a) The modeling is limited to the three topology templates. (b) The input sketch

lines will be analyzed as one template. (c) There are two parameters to control the flexibility: angle threshold and distance threshold. Angle threshold is used to evaluate the parallel information, while distance threshold is used to judge whether the strokes are meeting at one vertex or not.

- (2) Matching problem. This problem occurs very often for the beginners. However, it can be improved very quickly after several experiments. Furthermore, the system can show the edge graph on the right panel. This helps the user very much to correct the input sketch lines. The system supports unlimited UNDO operations.

8. Conclusion

We propose a sketch-based modeling system for the interactive modeling of a variety of 3D objects. The idea is to construct and modify 3D objects, step by step, using a 2D template topology library. The library, called T-LIB, consists of 2D Edge Graphs and the corresponding algorithms for modifying 3D objects. A local section of a 3D object is analyzed as a 2D Edge Graph. Once the Edge Graph can be matched to a template within T-LIB, the 3D object can be generated in a basic shape reconstruction procedure, or modified in 3D shape modification procedure. After we have obtained the polyhedral networks, we use fillet operation and subdivision to obtain a smooth shape. Future work involves adding more templates, the proposal of template definition rules and extending the system to be able to split larger unknown 2D Edge Graphs into smaller known sub-graphs.

References

- [Akeo*94] M. AKEO, H. HASHIMOTO, T. KOBAYASHI, T. SHIBUSAWA: Computer system for reproducing three-dimensional shape from idea sketch. In Eurographics'94 Proceedings, pp. 477-488, 1994.
- [Bloomenthal*98] M. BLOOMENTHAL, R.C. ZELEZNIK, R. FISH, L.S. HOLDEN, A.S. FORSBURG, R. RIESENFELD, M. CUTTS, S. DRAKE, H. FUCHS, AND E. COHEN: Sketch-N-Make: Automated Machining of CAD Sketches. Proceedings of the 1998 ASME 8th Computers In Engineering Conference, 1998.
- [Cherlin*2005] J. J. CHERLIN, F. SAMAVATI, M. C. SOUSA, J. A. JORGE: Sketch-based Modeling with Few Strokes. Proc. of the 21st Spring Conference on Computer Graphics (SCCG 2005), 2005, to appear.
- [Fukui*88] Y. FUKUI: Input Method of Boundary Solid by Sketching. Computer Aided Design, 20, 8, 434-440, 1988.

- [Furushima*90] S. FURUSHIMA, S. KANAI, H. TAKAHASHI: Generation of 3-Dimensional Geometric Model from Rough-sketch. NICOGRAPH No.6, pp. 11-20, 1990.
- [Hosaka*83] M. HOSAKA, F. KIMURA: Interactive Shape Design by Drawing Input. Proceedings of Computer Applications in Production and Engineering(CAPE 83), Amsterdam, pp. 1115-1125, 1983.
- [Igarashi*99] TAKEO IGARASHI, SATOSHI MATSUOKA, HIDEHIKO TANAKA: Teddy: A Sketching Interface for 3D Freeform Design. ACM SIGGRAPH'99, pp.409-416, 1999.
- [Kondo*88] K. KONDO, F. KIMURA, T. TAJIMA: Estimation of Viewport in Perspective Drawings and its Application (in Japanese). In J. Information Society of Japan, pp. 686-693, 1988.
- [Kuragano*2002] J. KURAGANO, Y. TAKARADA, H. SUZUKI AND F. KIMURA: Subdivision Surface Generation from a Set of Curves. IPSJ SIG Notes, 2002-CG-108, pp. 13-18, 2002.
- [Lamb*90] D. LAMB AND A. BANDOPADHAY: Interpreting a 3D Object From a Rough 2D Line Drawing. In ed. A.E.Kaufman, Proceedings of the First IEEE Conference on Visualization '90, 59-66, IEEE, 1990.
- [Liu*2003] WEIZHONG LIU, KUNIO KONDO: Adaptive Algorithm of Automatic Implementing Fillet Operations with Loop Subdivision Surfaces. SPIE, Vol. 4756, pp. 87-97, 2003.
- [Liu*2004] W. LIU, K. KONDO: An Adaptive Scheme for Subdivision Surfaces based on Triangular Meshes. Journal for Geometry and Graphics, Vol. 8, No. 1, pp. 69-80, 2004.
- [Matsuda*2003] K. MATSUDA, S. SUZUKI, K. KONDO: Sketch Interpreter System: 3D Modeling System with Handdrawn Shade and Shadow. The Transactions of the Institute of Electronics, Information and Communication Engineers, Vol.44, No.11, pp.2547-2555, 2003.
- [Mitani*2000] J. MITANI, H. SUZUKI, F. KIMURA: 3D Sketch: Sketch- Based Model Reconstruction and Rendering. IFIP Workshop Series on "Geometric Modeling: Fundamentals and Applications", 7th Workshop GEO-7, pp. 85-112, 2000.
- [Naya*2002] F. NAYA, J. A. JORGE, J. CONSESA, M. CONTERO, J. M. GOMIS: Direct Modeling: from Sketches to 3D Models, Proc. of the 1st Ibero-American Symposium in Computer Graphics, pp. 109-117, 2002.
- [Perkins*68] D. N. PERKINS: Cubic Corners. Quarterly Progress Report 89, 207-214, MIT Research Laboratory of Electronics, 1968.
- [Schweikardt*98] E. SCHWEIKARDT AND M. D. GROSS: Digital Clay: Deriving Digital Models from Freehand Sketches. ACADIA'98, 202-211.
- [Shesh*2004] A. SHESH AND B. CHEN: SMARTPAPER: An Interactive and User Friendly Sketching System. Computer Graphics Forum 23(3), 301-310, 2004.
- [Sugishita*94] S. SUGISHITA, K. KONDO, H. SATO, S. SHIMADA: Sketch Interpreter for geometric modeling. International Conf. of Computer Aided Geometric Design, 1994.7.
- [Varley*2000] P. A. C. VARLEY AND R. R. MARTIN: A System for Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch. Proc. GMP 2000, IEEE Press, 2000.
- [Varley*2002] P. A. C. VARLEY, H. SUZUKI, J. MITANI: R. R. Martin: Interpretation of Single Sketch Input for Mesh and Solid Models. International Journal of Shape Modeling, Vol. 6, No. 2(2002) 207-241.
- [Xu*95] Z. XU, S. SUGISHITA, K. KONDO, H. SATO, S. SHIMADA: Free-form Surfaces Generation in Freehand Sketch Interpreter System. CADEM'95, Xi'an, China, Oct.12-16, pp.261-266, 1995.
- [Zelevnik*96] ROBERT C. ZELEVNIK, KENNETH P. HERNDON, JOHN F. HUGHES: SKETCH: An Interface for Sketching 3D Scenes. SIGGRAPH'96, pp.163-170, 1996.