

## 面向概念设计的实体草绘建模<sup>\*</sup>

何利力<sup>1</sup>, 方贵盛<sup>2+</sup>, 孔繁胜<sup>2</sup>

<sup>1</sup>(浙江理工大学 信电学院, 浙江 杭州 310018)

<sup>2</sup>(浙江大学 计算机科学与技术学院, 浙江 杭州 310016)

### Sketch-Based Solid Modeling for Conceptual Design

HE Li-Li<sup>1</sup>, FANG Gui-Sheng<sup>2+</sup>, KONG Fan-Sheng<sup>2</sup>

<sup>1</sup>(College of Informatics and Electronics, Zhejiang Sci-Tech University, Hangzhou 310018, China)

<sup>2</sup>(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

+ Corresponding author: E-mail: fanggsh@mail.zjwchc.com

**He LL, Fang GS, Kong FS. Sketch-Based solid modeling for conceptual design. *Journal of Software*, 2008, 19(7):1817-1827. <http://www.jos.org.cn/1000-9825/19/1817.htm>**

**Abstract:** A simple, rapid and efficient approach to constructing 3D conceptual solid models on sketch-based user interface is presented. It emulates the traditional sketching design approach with paper and pencil to sketch strokes on the calligraphic interface, and then applies the feature-based modeling method to construct different sweep feature-based primitives according to the shape and location relationship of feature strokes. In order to construct complex solid models, the feature joining and cutting mechanism based on intention capturing is introduced. This paper illustrates the implementation of the algorithm, and experiment shows that the algorithm is efficient and well suitable for real time on-line 3D conceptual model creation.

**Key words:** conceptual design; sketch-based modeling; gesture; feature; Parasolid geometric kernel

**摘要:** 提出了一种简单、快速、高效的基于草绘设计的概念模型创建方法。该方法模仿传统的纸笔草绘设计方式,允许用户自由地在笔式用户界面上勾画特征笔划,然后根据笔划的形状和位置关系采用基于特征的实体建模方法构造不同的扫描特征体素。在此基础上,采用了基于意图捕捉的特征添加及切削机制创建复杂的实体模型,给出了一些关键算法的实现过程。实验结果表明,该方法能够较好地应用到联机三维概念模型快速创建过程中。

**关键词:** 概念设计;草绘建模;手势;特征;Parasolid 几何内核

中图法分类号: TP391 文献标识码: A

---

\* Supported by the National Basic Research Program of China under Grant No.2002CB312103 (国家重点基础研究发展计划(973)); the Young Teacher Foundation of Zhejiang Province of China under Grant No.200418 (浙江省高校青年教师基金)

Received 2006-12-28; Accepted 2007-03-28

## 1 Introduction

### 1.1 The importance of sketch-based 3D modeling in conceptual design

Conceptual design is the initial stage of product development which is characterized by inadequate and vague design information. However, current CAD systems need the input of precise dimensional and geometric constraints information in the product modeling process. At the same time, the use of WIMP-based user interface makes the modeling process very cumbersome, tedious and time-consuming. These reasons make current CAD systems mainly suitable for the detailed design but not the conceptual design. In the conceptual design stage, designers usually use pencil and paper to rapidly record their creative ideas. Only when the configuration of the product is determined fully, the digital model will be constructed by using the computer aided design technologies. Sketching is a simple means of expression and accessible by everyone. It is very useful for quickly recording the creative ideas and sharing them with the working team. It is also an excellent stimulant for creativity and innovation. So the design mode by sketching is widely used in the conceptual design stage of product development. But for the traditional sketching design mode by pen and paper, because the design process and the modeling process are separate, some important information may be lost during information conveyance. Currently, one of the key issues is how to directly transform the designer's ideas into the digital model which can be handled by CAD systems. Sketch-based 3D modeling user interface is user-centric, which provides a natural and efficient design environment similar to sketching by using pencil and paper. It integrates the advantages of both traditional sketching design style and computer aided design, and can utilize computers' powerful modeling and intelligent reasoning capabilities to rapidly express designers' brainstorm ideas.

### 1.2 Survey on Sketch-based 3D modeling methods

In this section we present an overview of related work by other researchers attempting to solve similar problems to the one outlined. We take an example, as shown in Fig.1, to elaborate various modeling methods by sketching strokes. There are some typical methods to construct a box solid model as shown in Fig.1(a).

Figure 1(b) shows the modeling approach based on gestures, which is used in SKETCH<sup>[1]</sup> and GIDeS<sup>[2]</sup>. It defines some basic gestures for instantiating primitives as sequences of strokes. For instance, a drawing of three non-collinear line segments which meet at a point imply a corner, consequently, we interpret similar gestures composed of three line strokes as a cuboid construction operation. Then the system analyses both the topology and geometry of gesture strokes and tries to construct a 3D primitive object.

Figure 1(c) shows the modeling approach by feature matching based on rules<sup>[3,4]</sup>. Different features have different inference rules. For examples, the box feature is composed of a closed profile and one extrusion line. Then the system uses the 2D connectivity information to determine the feature type. Once a specific feature type is found, the system utilizes all the necessary 3D information to produce the 3D feature.

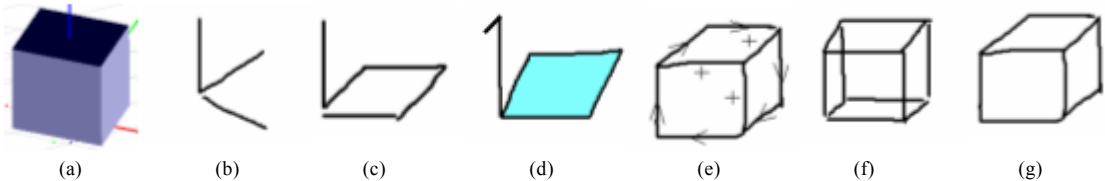
Figure 1(d) shows the modeling approach by creating extrusion feature<sup>[5]</sup>. The user sketches an arrow pen marking starting at a corner of a surface picked previously. Once the system interprets it as an extrusion command, the surface will move along the arrow pen marking to form a final 3D model.

Figure 1(e) shows the reconstruction-based modeling approach by using the Huffman-Clowes labeling scheme<sup>[6]</sup> which is an old machine vision algorithm. First the system recognizes the sketch as the line drawing, and then three kinds of line labeling which are '+', '-' and arrow, are placed on the drawing to deduce the initial topology of the object and to locate hidden faces.

Figure 1(f) shows the reconstruction-based modeling approach by using the optimization algorithm<sup>[7,8]</sup> or the axonometric inflation engine<sup>[9,10]</sup> or using human intuitions in shape recognition and human drawing convention<sup>[11]</sup>

to establish 3D models from freehand sketching strokes. The key technology of reconstruction-based modeling approach is how to compute the three dimensional coordinate values of every corner point in the sketch by reverse projection and then to construct the faces or bodies of the model.

Figure 1(g) shows the 3D modeling method based on 2D graph matching. Its basic idea is to create complex 3D objects with a template topology library step by step from freehand strokes. The input strokes are analyzed as a 2D edge graph. Once the edge graph can be matched to a template of the library, the 3D shape can be constructed<sup>[12]</sup>.



(a) Box object shown in shade mode; (b) Gestures-based modeling; (c, d) Feature-based modeling;  
 (e) Reconstruction-based modeling by using the Huffman-Clowes labeling scheme;  
 (f) Reconstruction-based modeling by using the optimization method; (g) Modeling method based on 2D graph matching

Fig.1 Several typical modeling methods to construct a box object by sketching strokes

For those systems mentioned above, we can find some main deficiencies:

- Poor modeling capability. Some systems can only create or recognize some simple 3D models such as boxes, cylinders, etc.
- The modeling process is relatively complex. Some systems need drawing many strokes in order to construct a simple 3D model.
- The systems are not robust enough for practical applications because they are mostly developed from scratch, and some current key CAD technologies are not well used in the sketch-based modeling process.

### 1.3 Our approach

The paper presents a kind of new modeling method by sketching strokes which uses the Parasolid as the modeling geometric kernel. Some typical solid models will be built by sketching two kinds of feature strokes in our design system: the profile stroke and the path stroke. The feature strokes can be a single primitive or a combined shape with lines and arcs. The system then creates suitable 3D solid models by means of the location relationship between the profile stroke and the path stroke. These models which can be constructed by creating the sweep feature in traditional CAD systems are very common in mechanical engineering domains, because some models created by extrusion feature or revolution feature can also be constructed by creating the sweep feature. In order to construct some complex solid models, the system uses the feature joining or cutting mechanism and supports users sketching strokes on the surface of the 3D solid model. Finally, the system uses the gesture-based method to edit solid models.

## 2 The Architecture for Sketch-Based 3D Conceptual Solid Modeling System

The proposed architecture for such sketch-based modeling system consists of an interaction layer, a recognition layer, a modeling layer, a kernel layer, and a rendering layer.

The interaction layer provides a sketch-based user interface which supports strokes input by using a pen or a mouse and sketching work plane selection.

The recognition layer classifies sketching strokes into the section stroke and the path stroke which are used to create the sweep feature and the gesture stroke for model editing. The section stroke and the path stroke are drawn with the left button of the mouse, and the gesture stroke is drawn with the right button of the mouse.

The modeling layer creates different sweep features in terms of the shape and the location relationship between the section stroke and the path stroke. The system supports sketching strokes on surfaces of solid models and features joining and cutting by using the intention capturing mechanism, and model editing based on gestures.

The kernel layer consists of the Parasolid kernel and the OpenGL tool kit. Parasolid provides powerful modeling functions to create complex feature-based solid models. OpenGL is used to display graphical objects on the screen.

The rendering layer performs the post-processing and visualization process. Users can rotate, zoom into or out from, or pan the view respectively by using the mouse button. At the same time, the system provides five kinds of modes to visualize models, and they are the shaded mode, the wireframe mode, the wireframe with silhouettes mode, the hidden line mode, and the gray hidden line mode.

### 3 Key Algorithms

In this section we will illustrate the details of several key algorithms developed for creating complex solid models by sketching strokes.

#### 3.1 The algorithm for recognizing two kinds of feature strokes

When a user draws directly a stroke on the screen with a pen or a mouse, the stroke is sampled as a series of points  $P_i$  ( $i=0,1,\dots,n$ ) according to time sequence. In order to create the regular solid model, we need to process the raw input stroke by capturing users' design intention. The process of strokes recognition and beautification is divided into five parts of strokes preprocessing, key points detection, primitives recognition, primitives reconstruction, primitives beautification.

##### 3.1.1 Strokes preprocessing

The aim of strokes preprocessing is to eliminate the number of noisy points and redundant points by smoothing the raw stroke and to shorten the computing time. Because the preprocessing method by re-sampling the raw stroke based on the distance criterion<sup>[13]</sup> will lose some important information, in this paper we adopt the method as follows.

Input:  $P_{in} = \{P_i : (x_i, y_i) (i=1, 2, \dots, n)\}$ ;

Output:  $P_{out} = \{P'_j : (x_j, y_j) (j=1, 2, \dots, m, m = \text{int}(n/2) + 1)\}$ .

The algorithm:

Step 1: Given the input points set:  $P_{in}$ ;

Step 2: Let  $P'_1 = P_1$ ;

Step 3: Compute the coordinate values of each point in the output points set  $P_{out}$  by Eq.(1)

$$P'_j = (P_{2(j-1)} + P_{2j}) / 2, \quad j \neq 1, j \neq m \quad (1)$$

Step 4: Let  $P'_m = P_n$ , we will get the output point set  $P_{out}$ .

##### 3.1.2 Key point detection

Key point is the segmentation point of the stroke. It is usually the inflection point or the corner point of the stroke. The aim of key point detection is to segment the uni-stroke into several sub-segments by the features of the stroke. In this paper we use some static or dynamic features of the sketching stroke such as the variety of the speed and the variety of the angle to detect key points.

Input:  $Q_{in} = \{P'_j : (x_j, y_j) (j=1, 2, \dots, m)\}$ ;

Output:  $Q_{out} = \{P''_k : (x_k, y_k) (k=1, 2, \dots, l)\}$ .

The algorithm:

- Step 1: Input the coordinates of every point of the set  $Q_{in}$ ;
- Step 2: Initialize some thresholds such as  $A, L, C$ , where  $A$  is the threshold of the angle,  $L$  is the threshold of the sketching speed, and  $C$  is the threshold of the curvature;
- Step 3: Compute the angle value  $\alpha_j$  between two adjacent vectors as shown in Fig.2(a) by Eqs.(2)~(4), where  $d_j$  is the distance of two adjacent points, and  $d'_j$  is the distance between  $P'_{j+2}$  and  $P'_j$ ;

$$d_j = |P'_{j+1} - P'_j| \tag{2}$$

$$d'_j = |P'_{j+2} - P'_j| \tag{3}$$

$$\alpha_j = \angle P'_j P'_{j+1} P'_{j+2} = \arccos\left(\frac{d_j^2 + d_{j+1}^2 - d'^2_j}{2d_j d_{j+1}}\right) \tag{4}$$

- Step 4: Compute the sketching velocity of the  $j$ -th point by Eqs.(5) and (6), where  $t$  is the average sampling time

$$\Delta d_j = d_{j+1} - d_j \tag{5}$$

$$v_j = \frac{\Delta d_j}{t} \approx \Delta d_j \quad (t \text{ is a constant value}) \tag{6}$$

- Step 5: Compute the variety value of curvature between two adjacent points  $\Delta c_j$  by Eqs. (7) and (8) as shown in Fig.2(b), where  $c_j$  is the curvature of the  $j$ -th point,  $\beta_j$  is the angle between the vector  $\overline{P_j P_{j+1}}$  and the X-axis;

$$c_j = \frac{\beta_{j-1} - \beta_j}{d_j}, \text{ where } \beta_j = \arctg \frac{y \times P'_{j+1} - y \times P'_j}{x \times P'_{j+1} - x \times P'_j} \tag{7}$$

$$\Delta c_j = c_{j+1} - c_j \tag{8}$$

- Step 6: If  $\alpha_j < A$ , the point  $P_{j+1}$  is the segmentation point, else go to Step 7;
- Step 7: If  $170 < \alpha_j < 180$ , and  $d < L$ , and  $|c_j| \leq C$ , the point  $P_j$  is also a key point;
- Step 8: Output the key point set:  $Q_{out}$ .

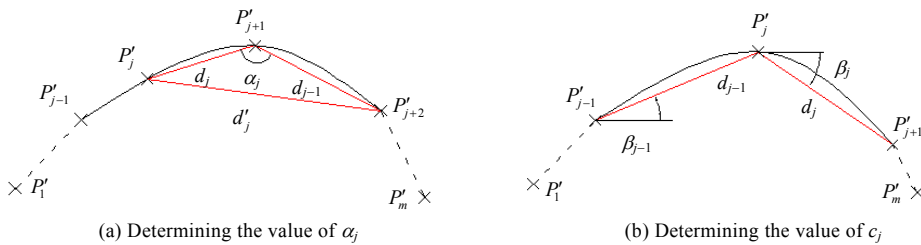


Fig.2 Detection of key points

### 3.1.3 Primitives recognition

After segmenting the stroke, we need to recognize the shape of every sub-segment in order to correctly capture users' sketching intention. A kind of method is presented to recognize the primitives as follows.

- Step 1: Given the input point set:  $P_{out}$  and  $Q_{out}$ ;
- Step 2: Compute the distance  $h$  between the point  $P'_{j+1}$  and the line connected by two segmentation points  $P''_{k+1}$  and  $P''_k$  in each sub-segment;
- Step 3: Find the maximum of  $h$ :  $h_{max}$  in each sub-segment;
- Step 4: Compute the value of  $\phi$  by Eq.(9). If  $\phi \leq 0.02$ , then the shape of the sub-segment is a line, else if  $\phi \leq 1.17$  and  $\phi > 0.02$ , then the shape of the sub-segment is an arc, else if  $\phi > 1.17$ , then the shape of the

sub-segment is a circle;

$$\phi = \frac{h_{\max}}{P_k'' - P_{k+1}''} \quad (9)$$

Step 5: Compute the feature parameters of every primitive. For the line, we take the start point and the end point of each sub-segment as the feature points. For circles and arcs, we use the least square fitting method to compute the radius and the coordinate value of the center point of the circle or the arc.

### 3.1.4 Primitive reconstruction and beautification

Primitive reconstruction uses the functions of Parasolid geometric kernel to reconstruct the geometric primitives by the shape of each sub-segment and their feature parameters.

After the reconstruction, the reconstructed primitives do not connect with each other sometimes and do not satisfy users' design intention. So we need to further beautify the primitives by using geometric constraint relations, for example, making the single line vertical or horizontal, making two lines have equal length or connect with each other, making two arcs have equal radius, etc.

Our rule-based primitive beautification algorithm can be described briefly in three steps:

Step 1: Connectivity constraints. The system looks at the end points of a pair of primitives, and determines whether they are within a certain distance tolerance. If they are, the two end points of the two primitives will be connected by using the rotation and movement translations;

Step 2: Horizontality and verticality constraints. If the angle between the line and the  $X$ -axis is within a threshold, the end points of the line will be adjusted to make one horizontal or vertical;

Step 3: Equivalence constraints. For the lines, if  $\varphi = \left| \frac{L_{pre} - L_{pro}}{L_{pre}} \right|$  is within a threshold, where  $L_{pre}$  is the length of the preceding reconstructed line,  $L_{pro}$  is the length of the subsequent reconstructed line, the system will adjust the end points of the subsequent reconstructed line to make them equal.

## 3.2 The mechanism for creating the sweep feature based on Parasolid

### 3.2.1 The algorithm for locating the approximate center of the close section

The computation of the approximate center of the close section is to determine the relative location relationship between the sweep path and the sweep section and to create various sweep features. Although some algorithms have been presented<sup>[14]</sup>, they are not suitable for on-line sketching design. A new algorithm for locating the approximate center of the close section is presented in this paper, in order to reduce the computational complexity.

Step 1: Input the key point set  $Q_{out}$ ;

Step 2: Find out the maximum and minimum coordinate values  $X_{\min}$ ,  $X_{\max}$ ,  $Y_{\min}$ ,  $Y_{\max}$  from the key point set  $Q_{out}$ ;

Step 3: Construct a rectangle by two diagonal points  $(X_{\min}, Y_{\min})$  and  $(X_{\max}, Y_{\max})$ ;

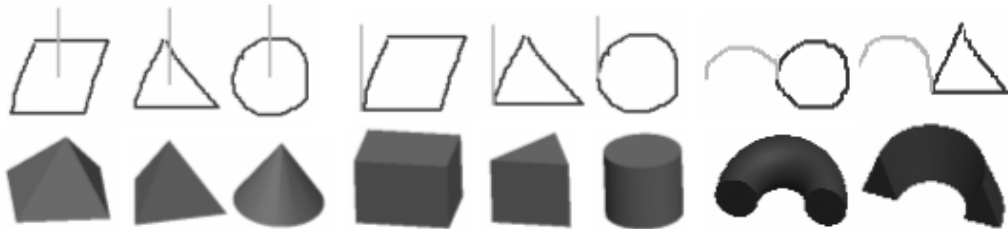
Step 4: Connect the diagonal lines of the rectangle and take the point of intersection as the approximate center of the close section.

If the key point set  $Q_{out}$  has  $m$  key points, and each key point has two coordinate values, then the time complexity is  $O(\lfloor \log m \rfloor)$  in step 2 in order to find out the maximum and the minimum of each coordinate value by using the binary search algorithm. Compared with the time complexity  $O(m)$  of the algorithm used in Ref.[14], our algorithm needs fewer time.

### 3.2.2 The algorithm for creating the sweep feature

Parasolid provides many powerful functions to create complex solid models. In this paper we use the sweep feature creation function: `PK_BODY_make_swept_body` to create the feature-based solid models. By setting

different feature parameters of the function, we can create not only the sweep features, but also the extrusion features and the revolution features. The extrusion feature can be looked as sweeping the section along the line, and the revolution feature can also be looked as sweeping the section along the arc or the circle as shown in Fig.3.



(The black stroke is the section stroke and the gray stroke is the path stroke)

Fig.3 Examples for creating various sweep-based feature solid models

Firstly, users should sketch the section stroke, and then sketch the path stroke. When the two kinds of strokes have been processed, the system creates different sweep features by strokes' location relationship. If the start point of the path stroke locates in a tolerance circle of the approximate center of the section stroke, the system will create the sweep feature with variable sections. Else the system will create the sweep feature with invariable section.

The algorithm for creating the sweep feature is described as follows:

Step 1: The user sketches the section stroke;

Step 2: The system segments, recognizes and reconstructs the section stroke;

Step 3: If the reconstructed stroke is close, then the system sets  $Tag=1$  and computes the approximate center of the section and prompts the user to sketch the path stroke, go to Step 4; Else the system sets  $Tag=0$  and prompts the user to redraw the section stroke, go to Step 1;

Step 4: The user sketches the path stroke, and then the system processes it using the same method as that for the section stroke;

Step 5: If the start point of the path stroke locates in a tolerance circle of the approximate center of the section stroke, the system will create the sweep feature with variable sections. Else the system will create the sweep feature with invariable section;

Step 6: The system sets  $Tag=0$  and prompts the user to draw the section stroke again, then go to Step 1.

### 3.3 The algorithm for features joining and cutting operation

In order to construct complex models, the system supports the user sketching over an existent object in the scene, the newly created feature is properly placed and attached to the object, and the system tries to automatically identify the joining or cutting operation of the feature.

#### 3.3.1 Selection of the sketching work plane

In order to sketch the profile and path over the surface of the model, the user should rotate the surface to the current viewpoint, then presses the right button of the mouse or the assistant key of the stylus on the suitable position of the surface. The algorithm about selection of the sketching work plane is described as follows:

Step 1: Get the location of the mouse and translates the screen coordinate value of the point into the world coordinate value;

Step 2: Construct a ray to parallel to the orientation of the current viewpoint which starts from the point of the mouse;

Step 3: Search parts in the scene. If the part is not empty, then we find all surfaces of the part which intersect

with the ray. Finally, we take the surface of the part which has the shortest distance with the start point of the ray as the sketching work plane.

### 3.3.2 The discrimination mechanism of joining and cutting operations of features

Users usually express the design intention by using some special strokes combination during the process of conceptual design. For example, in Fig.4(a), when the user sketches feature strokes aside a existent model, he means to increase a sweep-based feature. In Fig.4(b), when the user sketches feature strokes inside an existent model, he means to cut a hole from the existent model. In Fig.4(c), when the user sketches feature strokes on the surface of the existent model, he also means to increase a sweep-based feature. But in Fig.4(d), Fig.4(e), if the feature strokes intersect with the existent model, it will have two possibilities. One is feature joining, the other is feature cutting. The system should distinguish them correctly.

In this paper we describe a method to automatically discriminate joining and cutting operations of features. The system computes the intersection set between the newly created feature and the existent model. If the intersection set is empty, the newly created feature is united with the existent model as shown in Fig.4(a). If the intersection set is the newly created feature, then the feature is subtracted from the existent model as shown in Fig.4(b). If the intersection set is a facet, the system also joins the newly created feature with the existent model as shown in Fig.4(c). In other instance, there are two kinds of probabilities. The system uses the context sensitive menu to resolve ambiguity as shown in Fig.4(d), Fig.4(e), you can choose the “cut” icon or the “join” icon to create more complex models.

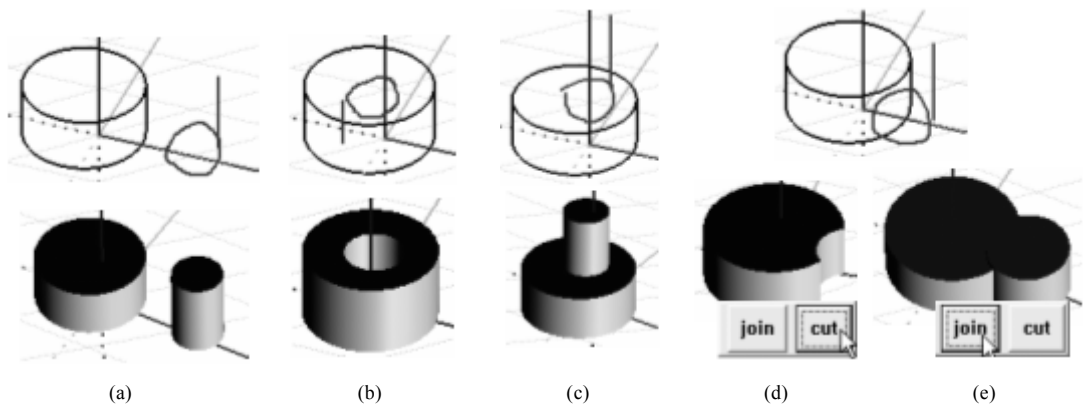


Fig.4 Automatically inferred feature joining and cutting operations (a, b, c) and the context sensitive menu for ambiguous resolution (d, e)

## 4 Implementation and Preliminary Evaluation

Our interactive sketching modeling system, Sketch3D has been implemented on Window XP environment using Visual C++ 6.0. The Parasolid is used as the geometry kernel to create feature-based solid models, the OpenGL graphics library is used to display graphical objects.

### 4.1 Feature stroke sketching and recognition

That the sketching strokes are recognized correctly by the system is the key problem which insures that the modeling result accords with users' design intention. In order to test the result of the algorithms mentioned above, the author sketches some feature strokes in the system environment as shown in Fig.5. The upper row in Fig.5 is the raw sketching strokes, the lower row is the recognition strokes. From Fig.5, we can see, the raw sketching strokes



can be recognized and reconstructed correctly, and the reconstructed primitives can be connected with each other to form a close profile section.

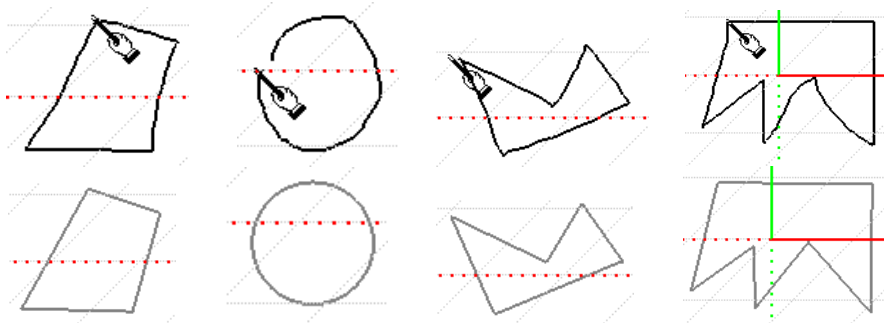


Fig.5 Some examples on the feature strokes sketching and recognition

#### 4.2 The 3D modeling result and preliminary evaluation

Figure 6 shows some examples of the solid models generated with our system. The execution time needed for an experienced user using a commercial CAD system is also provided.

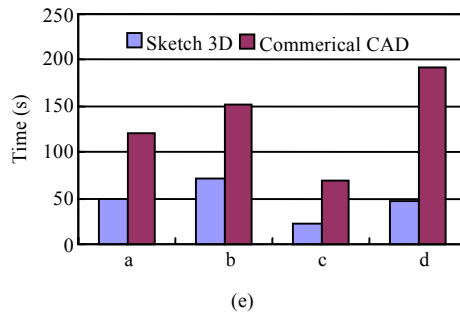
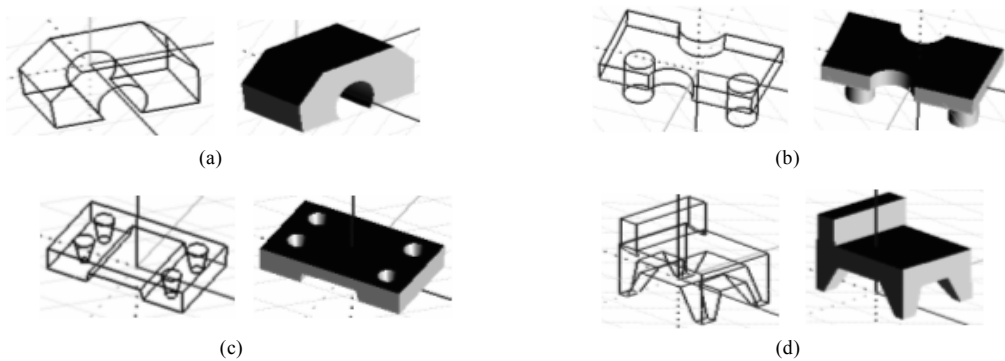


Fig.6 Some examples created by our system and the elapsed time to draw these models

From Fig.6, we can see that it costs less time to create the same model using our system than using the commercial CAD system. Moreover, the number of commands needed to create the same models is smaller in our system.

Compared with the sketching modeling methods mentioned in Section 2, our approach can create not only some basic shapes such as boxes, cylinders, but also some complex solid models. Furthermore, the modeling process is very simple, the user needs drawing few strokes in order to construct a 3D conceptual model. The third

advantage of the system is the realization of the integration of the sketching design method and the Parasolid geometric kernel which ensures that the modeling result in our system can be used in other CAD systems.

## 5 Conclusions and Future Work

This paper has presented some basic ideas and algorithms to progressively create and edit 3D conceptual models by using hand drawn sketches. These algorithms have been used in a sketch-based feature modeling prototype system, dubbed Sketch3D, for conceptual design.

Although some elementary research work has been done in order to improve the modeling ability of the current CAD systems in the stages of conceptual design, there are still some limitations which we need further improvement. At present, in the stage of 2D reconstruction we classify the input stroke into some elementary primitives, such as line segments, circles, and arcs, or gesture commands. This will be extended in the near future to induce other primitives, such as ellipses and spline curves. At the same time, we will explore the creation mechanism of the blend feature by sketching strokes. Once the user finishes sketching two profiles, the blend-based feature model will be created automatically.

### References:

- [1] Zeleznik RC, Herndon KP, Hughes JF. SKETCH: An interface for sketching 3D scenes. In: Computer Graphics Proc., Proc. of the 23rd Annual Conf. Series. ACM SIGGRAPH. New York: ACM Press, 1996. 161–170.
- [2] Pereira JP, Jorge JA, Branco VA, Ferreira FN. Calligraphic interfaces: Mixed metaphors for design. LNCS 2844, Berlin: Springer-Verlag, 2003. 154–170.
- [3] Qin SF, Wright DK, Jordanov IN. From on-line Sketching to 2D and 3D geometry: A system based on fuzzy knowledge. Computer-Aided Design, 2000,32(14):851–866.
- [4] Ma CX, Zhang FJ, Chen YD, Dai GZ. Feature-Gestures modeling of sketch in conceptual design. Journal of Computer-Aided Design & Computer Graphics, 2004,16(4):559–565 (in Chinese with English abstract).
- [5] Kim DH, Kim MJ. A new modeling interface for the pen-input displays. Computer-Aided Design, 2006,38(3):210–223.
- [6] Schweikardt E, Gross MD. Digital clay: Deriving digital models from freehand sketches. Automation in Construction, 2000,9(1): 107–115.
- [7] Masry M, Kang DJ, Lipson H. A freehand sketching interface for progressive construction and analysis of 3D objects. Journal of Computers & Graphics, 2005,29(4):563–575.
- [8] Lipson H, Shpitalni M. Correlation-Based reconstruction of a 3D object from a single freehand sketch. In: Proc. of the AAAI Spring Symp. on Sketch Understanding. Cambridge: AAAI Press, 2002. 99–104.
- [9] Naya F, Conesa J, Contero M, Company P, Jorge J. Smart sketch system for 3D reconstruction based modeling. LNCS 2733, Berlin: Springer-Verlag, 2003. 58–68.
- [10] Contero M, Naya F, Jorge J, Contero M, Conesa J. CIGRO: A minimal instruction set calligraphic interface for sketch-based modeling. LNCS 2699, Berlin: Springer-Verlag, 2003. 549–558.
- [11] Shesh A, Chen BQ. SMARTPAPER: An interactive and user friendly sketching system. Computer Graphics Forum, 2004,23(3): 301–310.
- [12] Liu WZ, Kondo K, Mitani J. A freehand sketch interpreter system for constructing 3D solid models. 2005. <http://www.interaction-ipsj.org/archives/paper2005/pdf2005/interactive/B201.pdf>
- [13] Kim DH, Kim MJ. A curvature estimation for pen input segmentation in sketch-based modeling. Computer-Aided Design, 2006, 38(3):210–223.
- [14] Chen T, Ai TH. Automatic extraction of skeleton and center of area feature. Geomatics and Information Science of Wuhan University, 2004,29(5):443–446 (in Chinese with English abstract).

## 附中文参考文献:

- [4] 马翠霞,张风军,陈由迪,戴国忠.支持概念设计的特征手势建模.计算机辅助设计与图形学学报,2004,16(4):559-565.  
 [14] 陈涛,艾廷华.多边形骨架线与形心自动搜寻算法研究.武汉大学学报(信息科学版),2004,29(5):443-446.



**HE Li-Li** was born in 1966. He is a professor at the Zhejiang Sci-Tech University. His research areas are CIMS and GIS.



**KONG Fan-Sheng** was born in 1946. He is a professor and doctoral supervisor at the Zhejiang University. His research areas are AI and CAD/CG.



**FANG Gui-Sheng** was born in 1973. He is a Ph.D. candidate at the Zhejiang University. His research areas are CAD/CAM and Sketch-based modeling design.

\*\*\*\*\*

### CALL FOR PAPERS

#### 2008 IEEE International Symposium on IT in Medicine & Education (ITME 2008)

12-14 Dec., 2008, Xiamen, China

<http://itme.xmu.edu.cn/>

IT changes everyday's life, especially in education and medicine. The goal of 2008 IEEE International Symposium on IT in Medicine & Education (ITME 2008) is to further explore the theoretical and practical issues of IT in education and medicine. It also aims to foster new ideas and collaboration between researchers and practitioners. The organizing committee is soliciting unpublished papers for the main conference and its special tracks. More information, please visit <http://itme.xmu.edu.cn/>.

Papers reporting original research results and experience are solicited. Each paper, written in English, should be no more than 6 pages (IEEE proceedings format), including references and Illustrations. E-submissions in PDF format are strongly recommended. Submission of a paper should be regarded as an undertaking that, should the paper be accepted, at least one of the authors will attend the conference to present the paper.

All submitted papers will be reviewed by at least two program committee members and will be accepted based on their originality, significance, relevance, and clarity of presentation. Accepted papers will be published in the IEEE categorized conference proceedings(will be indexed by EI & ISTEP). Authors of selected papers will be invited to submit revised and expanded version of their papers to be considered for a further publication in special issues of well-known Chinese or English journals cited by SCI or EI.

#### Contact information

Ms. Suxia Xu

Department of Cognitive Science, Xiamen University, China, 361005

Phone: +86-592-2580168, Fax: +86-592-2580258, E-mail: [suxiaxu@xmu.edu.cn](mailto:suxiaxu@xmu.edu.cn), [itme@xmu.edu.cn](mailto:itme@xmu.edu.cn)

#### Important Dates

- Paper submissions: Jul. 15, 2008
- Notification of acceptance: Aug. 15, 2008
- Camera-ready paper deadline: Sep. 15, 2008
- Early-bird registration: Sep. 15, 2008