Sketching in the Air: A Vision-Based System for 3D Object Design

Yu Chen[1], Jianzhuang Liu[1]
[1]Department of Information Engineering
The Chinese University of Hong Kong
{ychen6,jzliu}@ie.cuhk.edu.hk

Xiaoou Tang[1,2]
[2]Microsoft Research Asia
Beijing, China
xtang@microsoft.com

Abstract

3D object design has many applications including flexible 3D sketch input in CAD, computer game, webpage content design, image based object modeling, and 3D object retrieval. Most current 3D object design tools work on a 2D drawing plane such as computer screen or tablet, which is often inflexible with one dimension lost. On the other hand, virtual reality based methods have the drawbacks that there are awkward devices worn by the user and the virtual environment systems are expensive. In this paper, we propose a novel vision-based approach to 3D object design. Our system consists of a PC, a camera, and a mirror. We use the camera and mirror to track a wand so that the user can design 3D objects by sketching in 3D free space directly without having to wear any cumbersome devices. A number of new techniques are developed for working in this system, including input of object wireframes, gestures for editing and drawing objects, and optimization-based planar and curved surface generation. Our system provides designers a new user interface for designing 3D objects conveniently.

1. Introduction

Despite great progress of 3D modeling in current computer-aided design (CAD) tools, creating 3D objects using these tools is still a tedious job since they require users to work on a 2D drawing plane. Design in virtual 3D environments enables users to draw objects in 3D space, but this method has the drawbacks that there are awkward devices worn by the user and the virtual environments are expensive.

In this paper, we propose a novel vision-based approach to 3D object design. Different from the current techniques, it works in 3D space without any devices connected to the user. Our target is to develop an inexpensive system that allows the user to design 3D objects conveniently. Our system consists of a PC, a camera, and a mirror, as shown in Fig. 1. In the system, the user designs 3D objects by sketching in the air the wireframes of the objects with an easily-tracked wand. A number of sketching and editing operations are developed to facilitate object design. In real time, the 3D positions of the strokes of the wand are captured, and the wireframes and surfaces being developed are displayed on the PC screen to guide the user to draw more and more complex objects.

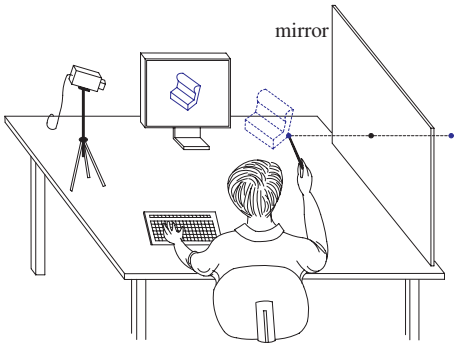

Fig. 1. The sketching system.

The system provides a whole new way of 3D object design. It requires no special equipments and is easy to set up and use. Its applications include flexible 3D sketch input in CAD, game, education, and webpage content design, generation of 3D objects from 2D images, and a user-friendly query interface for 3D object retrieval.

2. Related Work

Great effort has been made to develop CAD systems for 3D model design in the past three decades. Current techniques can be classified into the following four categories:

1) Traditional CAD tools such as AutoCAD [1] and SolidWorks [2]. These tools are sophisticated systems suitable for engineers to input precise geometry of models but are not suitable for designers to rapidly express their ideas at the initial stage of model development.

2) Automatic 3D object reconstruction from 2D line drawings. This is one of the main research topics in computer vision and graphics. The methods are mainly based on line labeling, algebra, image regularities, and optimization [18], [11], [17], [12], [5]. The critical problem in these methods is that they can handle only relatively simple pla-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

nar objects at the current stage.

3) Sketch-based modeling user interfaces. Most traditional designers still prefer pencil and paper to mouse and keyboard in current CAD systems to sketch their ideas of shapes. To bridge the gap between the flexible 2D sketches and the rigid CAD systems, researchers have developed tools that try to convert 2D sketches into 3D models [17], [19], [8], [3], [9]. However, one physical limitation that cannot be overcome by these tools is that the sketching and editing operations are performed on a 2D plane (tablet or screen). With one dimension missing, the 3D positions of the strokes, surfaces, and objects drawn on a 2D plane are often ambiguous.

4) Design in virtual 3D environments. Virtual reality (VR) has been thought to be the perfect CAD system because the designer could work naturally and intuitively in a real 3D environment. However, such systems face problems in the cost of the equipments, the inflexibility to use, and the slow frame update rates. Researchers are trying to develop better techniques for 3D design in VR [16], [10], [4], [7] but they need special and awkward devices to operate by, or connect to, the user, making the design an unnatural process.

From the discussion above, we can see that the current methods for 3D model design are not good enough. Researchers still need to develop more friendly and inexpensive interfaces with better design methodology.

3. The Sketching System

As shown in Fig. 1, our system consists of a video camera, a mirror, and a PC only. The user draws an object with a wand in the 3D free space. The tip of the wand is colored so that it is easy to track. The basic idea of 3D design in this system is that a 3D wireframe of an object is obtained by tracking the movement of the wand in 3D space, and then an automatic filling-in process generates a surface from the wireframe. We propose this system based on the observation that a designer thinks not in terms of surfaces, but rather in terms of the feature curves of an object which construct the wireframe. The whole flow chart of our system is shown in Fig. 2.

3.1. 3D Geometry of the System

Being able to find the 3D position of the tip of the wand is the first step. In the system, the world frame (X, Y, Z) is defined as in Fig. 3, where the XY plane is parallel to the image plane xy of the camera and the distance between the origin $\mathbf{O}$ and the image plane is equal to the focal length f. With a simple calibration, the YZ plane can be set orthogonal to the mirror. The angle θ between the Z axis and the mirror is less than 90 degrees so that the tip of the wand $\mathbf{P}_1 = (X_1, Y_1, Z_1)^T$ and its image $\mathbf{P}_2 = (X_2, Y_2, Z_2)^T$ in the mirror always project to two

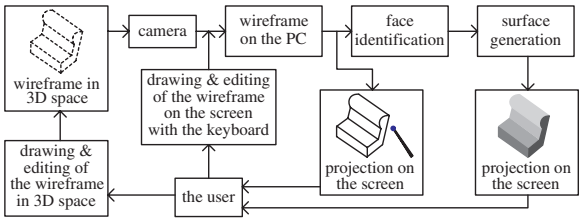

Fig. 2. Flow chart of 3D object design in the sketching system.

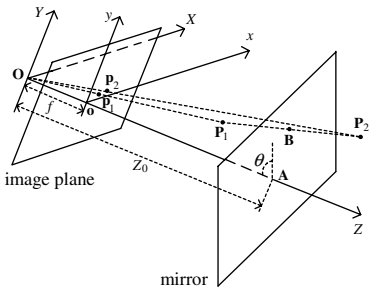

Fig. 3. Geometry of the system.

different points $\mathbf{p}_1 = (x_1, y_1, f)^T$ and $\mathbf{p}_2 = (x_2, y_2, f)^T$ on the image plane. To find the 3D coordinate of $\mathbf{P}_1$, we need to know θ and Z_0, where Z_0 is the distance from $\mathbf{O}$ to $\mathbf{A}$ and $\mathbf{A}$ is the intersection of the Z axis and the mirror. The two parameters θ and Z_0 can be obtained by the calibration scheme discussed in Section 4.2.

The 3D position $\mathbf{P}_1 = (X_1, Y_1, Z_1)^T$ can be determined from the known θ, Z_0, $\mathbf{p}_1$, $\mathbf{p}_2$, and the geometrical relationship shown in Fig. 3. The formula for calculating $\mathbf{P}_1$ is derived in Section 4.1.

Our system is able to determine the 3D positions of the wand with sufficient accuracy. We have also tested the traditional stereo method using two cameras to find the depth of a spatial point. From our experiments, we have found that the new method has the following advantages: (a) easier to calibrate, (b) less time to track the wand due to only one video sequence to handle, and (c) larger 3D working space for object drawing if the volume to set up the two systems are the same. The last advantage comes from the fact that if the traditional stereo method is used, the tip of the wand must appear in both image sequences of the cameras, which limits the 3D drawing space.

3.2. Locating the Wand

Locating the 3D position $\mathbf{P}_1$ of the wand is the first step for the system to work. We can represent $\mathbf{P}_1 = (X_1, Y_1, Z_1)^T$ in terms of the points $\mathbf{p}_1$ and $\mathbf{p}_2$ on the image plane, and the parameters f, θ, and Z_0. First, from the geometrical relationship in Fig. 3, it is straightforward to relate the spatial positions with the positions in the image plane by

$$X_i = x_i Z_i / f, \quad Y_i = y_i Z_i / f, \quad i = 1, 2. \tag{1}$$

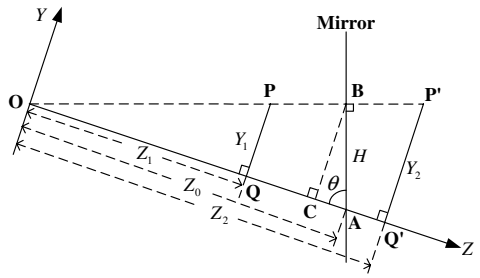

Fig. 4. Geometry of the system on the YZ plane.

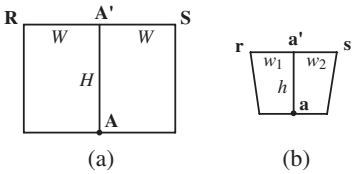

Fig. 5. (a) A rectangle for calibration. (b) The rectangle in the image.

We now consider the geometry of the system on the YZ plane as shown in Fig. 4, where $\mathbf{P}$ and $\mathbf{P}'$ are the projections of $\mathbf{P}_1$ and $\mathbf{P}_2$ onto the YZ plane, respectively. Let $\mathbf{B}$ be the midpoint of $\mathbf{PP}'$, and the three lines $\mathbf{PQ}$, $\mathbf{P}'\mathbf{Q}'$, and $\mathbf{BC}$ be perpendicular to the axis $\mathbf{OZ}$. Then $|\mathbf{PQ}| + |\mathbf{P}'\mathbf{Q}'| = 2|\mathbf{BC}|$ and $|\mathbf{OQ}| + |\mathbf{OQ}'| = 2|\mathbf{OC}|$. Hence, we have

$$\begin{aligned} Y_2 &= |\mathbf{P}'\mathbf{Q}'| = 2|\mathbf{BC}| - |\mathbf{PQ}| \\ &= 2H\sin\theta - y_1 Z_1/f, \qquad (2) \\ Z_2 &= |\mathbf{OQ}'| = 2|\mathbf{OC}| - |\mathbf{OQ}| \\ &= 2(Z_0 - H\cos\theta) - Z_1. \qquad (3) \end{aligned}$$

On the other hand,

$$\begin{aligned} H &= |\mathbf{AB}| = |\mathbf{PQ}|\sin\theta + |\mathbf{AQ}|\cos\theta \\ &= y_1 Z_1 \sin\theta/f - (Z_1 - Z_0)\cos\theta. \qquad (4) \end{aligned}$$

From (1), (2), (3), and (4), we have,

$$\begin{aligned} \frac{y_2}{f} &= \frac{Y_2}{Z_2} = \frac{2H\sin\theta - y_1 Z_1/f}{2(Z_0 - H\cos\theta) - Z_1} \\ &= \frac{2y_1 Z_1 \sin^2\theta - (Z_1 - Z_0) f \sin 2\theta - y_1 Z_1}{2Z_0 f \sin^2\theta - y_1 Z_1 \sin 2\theta + Z_1 f \cos 2\theta}. \qquad (5) \end{aligned}$$

Finally, Z_1 is obtained from (5) as

$$Z_1 = \frac{2fZ_0\sin\theta(y_2\sin\theta - f\cos\theta)}{(y_1 y_2 - f^2)\sin 2\theta - f(y_1 + y_2)\cos 2\theta}. \qquad (6)$$

Given (1), we immediately have the other two dimensions of $\mathbf{P_1}$: $X_1 = x_1 Z_1/f$ and $Y_1 = y_1 Z_1/f$. The position of the wand $\mathbf{P_1}$ in the 3D space is hence determined if f, θ, and Z_0 are known.

3.3. Calibration

The calibration process is to find the parameters θ and Z_0. It is reasonable to assume that the Z axis in Fig. 3 is orthogonal to the image plane and passes through the center $\mathbf{o}$ of the image displayed on the screen. The focal length f can be known from the camera and is fixed in the system. To calibrate the system, we print out a rectangle (Fig. 5(a)) on a white page and place this page on the central part of the mirror with the side $\mathbf{RS}$ approximately parallel to the ground. This rectangle is captured by the camera and displayed on the screen as shown in Fig. 5(b). Then we adjust the position of the camera so that the point $\mathbf{a}$ is coincident with the image center $\mathbf{o}$, the side $\mathbf{rs}$ is horizontal, and $w_1 = w_2$ in the image. This simple adjustment makes the YZ plane orthogonal to the mirror. With the known lengths of the sides of the rectangle in Fig. 5(a) and h and w_1 in the image, we can find θ and Z_0 from

$$\frac{w_1}{W} = \frac{h}{H\sin\theta} = \frac{f}{Z_0 - H\cos\theta}, \qquad (7)$$

which is derived from the geometry shown in Fig. 6.

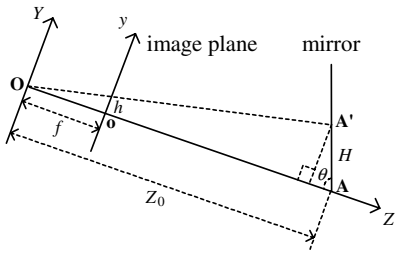

Fig. 6. Geometry of the YZ plane in calibration.

4. Wireframe Input and Object Editing

One difficulty to generate a wireframe of an object lies in the fact that the strokes drawn are invisible to the user. However, they are visible to the camera, and the trace of the moving wand and what has been drawn can be displayed on the screen as the feedback to the user, which can be used to guide the sketching process.

First, we propose a solution to locating the position on an unfinished wireframe in order to continue to draw. While the user moves the wand in the space, the closest point on the unfinished wireframe to the tip is computed, and a different color is shown on the screen to indicate this point. In this way, the user can find a position to draw without difficulty. When this position is found, the user presses some key to let the system know it, and then the movement of the tip is considered as a new edge of the wireframe.

Second, to distinguish a drawing stroke from non-drawing movement of the wand, we use the keyboard to let the system know when a stroke begins and ends. Besides, we have also developed a number of sketching and editing operations to facilitate object design, such as extrusion, moving, copying, rotation, and zooming. These 3D operations and gestures are summarized in Table 1. The keyboard is used to control the start and stop of a 3D gesture shown

Table 1. Gestures and operations defined in the system.

Keyboard	Function
m/r	move/copy the selected part
Left/Right	rotate along Y-axis
Up/Down	rotate along X-axis
+/-	zoom in/out
d	mode 1: move the position of a vertex or a control point mode 2: sketch a patch by dragging a straight/curve edge mode 3: sketch a pyramid/cone structure by dragging a patch mode 4: sketch a rectangular/cylindrical volume by dragging a patch
a	draw a straight line/curve
c/e	mode 1: draw a circle/ellipse mode 2: draw a body of revolution from a straight/curve edge
q/w	adjust the scaling of the selected part
z/x	adjust the curvature of curved strokes
1-4	change editing modes
F1	switch between the curve mode and the straight-line mode

by the movement of the wand. All the operations can be done by the wand and the keyboard, without resorting to the mouse, thus allowing a continuous design by moving the wand with one hand and hitting the keyboard with another. The system also allows the user to switch between two drawing modes: the curve mode and the straight-line mode. In the curve-mode, smooth Bezier curves are generated to fit the path of wand movement when forming the wireframe. On the other hand, in the straight-line mode, the path information is discarded and only the start and the end positions of each stroke is used to generate straight lines.

Compared with traditional sketch-based editing operations on a 2D plane, many 3D operations have their advantages. For example, if we want to draw a duct by the extrusion of a closed circle along an arbitrary open curve (see Fig. 7), a 2D system will encounter two problems: (a) whether the closed curve is a circle or ellipse and its orientation in 3D space are unclear; (b) the 3D trail of the open curve is impossible to determine. However, these problems do not exist in our system.

5. Surface Generation

When we obtain the wireframe of an object in the 3D space by using the sketching scheme described in the previous section, the next step is to generate the 3D surface from the wireframe to finally reconstruct the object. Surface generation can be divided into two steps. The first is to identify all the faces, i.e., the circuits in the wireframe which represent patches constituting the whole surface, and the second

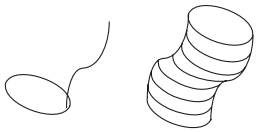

Fig. 7. Extrusion of the circle along the curve.

step is to generate these patches, either planar or curved, from their boundaries.

5.1. Face Identification

Given a wireframe, before filling in it with surface patches, we have to identify the circuits that represent these patches (faces). Since the wireframe may represent a manifold or non-manifold solid, a sheet of surface, or the combination of them, with or without holes, identifying the faces is not a trivial problem due to the combinatorial explosion in the number of circuits in the wireframe [13], [14], [15]. To solve this problem, we use the algorithm proposed in [15] to detect the faces of a wireframe. In our interactive system, the user can also select the edges of a face for face identification manually, which can help fix wrongly detected faces by the algorithm occasionally.

5.2. Planar Surface Generation

An object may have many planar faces. In the system, a straight line replaces a stroke in the straight-line mode. It is reasonable to consider that a face is planar if all its edges are straight lines. However, for a planar face with more than three vertices, it is not likely for all the vertices to be located exactly on a plane in 3D space due to the inaccuracy of the measurement and the input during the sketching process. Filling in these circuits with triangular patches will make the object distorted. In order to solve this problem, we propose an automatic line drawing correction algorithm to deal with this problem.

After face identification from a (partial) wireframe, we know the circuits representing planar faces. From the vertices of these circuits, a fitting algorithm is used to find a set of planes that best fit these planar circuits. We represent a plane passing through face j by its normal vector $\mathbf{f}_j = (a_j, b_j, c_j)^T$ and a scale d_j. Then, any vertex $\mathbf{v} = (x, y, z)^T$ on this plane satisfies the linear equation: $a_j x + b_j y + c_j z - d_j = 0$ or $\mathbf{v}^T \mathbf{f}_j = d_j$.

We hope that for each identified face, the corrected positions of its vertices should be as close to the fitting plane as possible. Besides, for each vertex, its corrected position should not deviate too much from its initial position. Let $\mathcal{V}_j$ be the set of the vertices of face j. The objective function to be minimized is defined as follows:

$$Q(\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N, \mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_M, d_1, d_2, \cdots, d_M) = \sum_{i=1}^{N} \|\mathbf{v}_i - \mathbf{v}'_i\|^2 + \beta \sum_{j=1}^{M} \sum_{i \in \mathcal{V}_j} \frac{\|\mathbf{v}_i^T \mathbf{f}_j - d_j\|^2}{\|\mathbf{f}_j\|^2}, \quad (8)$$

where $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N$ are the corrected positions of N vertices; $(\mathbf{f}_1, d_1), (\mathbf{f}_2, d_2), \cdots, (\mathbf{f}_M, d_M)$ are the parameters of M planar faces; $\mathbf{v}'_1, \mathbf{v}'_2, \cdots, \mathbf{v}'_N$ are the positions of the N vertices in the original sketching; β is a weighting factor. The goal of the optimization is to find the corrected positions $\mathbf{v}_i$, $i = 1, 2, \cdots, N$, and the fitting planes $(\mathbf{f}_j, d_j)$, $j = 1, 2, \cdots, M$, such that Q is minimized.

We solve this optimization problem in an iterative way. Let the set of $\mathbf{v}_i$, $i = 1, 2, \cdots, N$, the set of $\mathbf{f}_j$, $j = 1, 2, \cdots, M$, and the set of d_j, $j = 1, 2, \cdots, M$ be $V = \{\mathbf{v}_i\}_{i=1}^N$, $F = \{\mathbf{f}_j\}_{j=1}^M$, and $D = \{d_j\}_{j=1}^M$, respectively. Also let $V^n = \{\mathbf{v}_i^n\}_{i=1}^N$, $F^n = \{\mathbf{f}_j^n\}_{j=1}^M$, and $D^n = \{d_j^n\}_{j=1}^M$ be the optimization results after the nth iteration. The optimization problem is divided into two iterative minimization steps, and a closed-form solution can be achieved in each step.

Step 1: Face fitting.

$$(F^{n+1}, D^{n+1}) = \arg\min_{F,D} Q(V^n, F, D). \quad (9)$$

Step 2: Vertex correction.

$$V^0 = \{\mathbf{v}'_i\}_{i=1}^N, \quad (10)$$

$$V^{n+1} = \arg\min_{V} Q(V, F^{n+1}, D^{n+1}). \quad (11)$$

In Step 1, we do the plane fitting on all the identified planar faces using the updated positions of the vertices obtained in the previous iteration. The optimal fitting is obtained as follows. First, by solving $\frac{\partial Q}{\partial d_j} = 0$, we have

$$d_j = \frac{1}{|\mathcal{V}_j|} \sum_{i \in \mathcal{V}_j} \mathbf{v}_i{}^T \mathbf{f}_j, \quad j = 1, 2, \cdots, M. \quad (12)$$

Substituting (12) into (8), after some algebraic manipulation, we can transform the problem in (9) into the problem of minimizing the Rayleigh quotient $\frac{\mathbf{f}_j^T \mathbf{S}_j \mathbf{f}_j}{\mathbf{f}_j^T \mathbf{f}_j}$ with respect to each $\mathbf{f}_j$, $j = 1, 2, \cdots, M$, where $\mathbf{S}_j = \frac{1}{|\mathcal{V}_j|} \sum_{i \in \mathcal{V}_j} (\mathbf{v}_i - \overline{\mathbf{v}}_j)(\mathbf{v}_i - \overline{\mathbf{v}}_j)^T$ is the covariance matrix, and $\overline{\mathbf{v}}_j = \frac{1}{|\mathcal{V}_j|} \sum_{i \in \mathcal{V}_j} \mathbf{v}_i$. Furthermore, minimizing $\frac{\mathbf{f}_j^T \mathbf{S}_j \mathbf{f}_j}{\mathbf{f}_j^T \mathbf{f}_j}$ can be reduced to the following eigen-problem:

$$\mathbf{S}_j \mathbf{f}_j = \lambda_{j,min} \mathbf{f}_j, \quad j = 1, 2, \cdots, M, \quad (13)$$

with $\mathbf{f}_j$ being the eigen vector corresponding to the minimum eigen value $\lambda_{j,min}$ of $\mathbf{S}_j$. From (12) and (13), we can obtain the closed-form solution $\mathbf{f}_j^{n+1}$ and d_j^{n+1}, $j = 1, 2, \cdots, M$, in terms of $\mathbf{v}_i^n$, $i = 1, 2, \cdots, N$.

Step 2 is done by minimizing Q, given the fitting planes obtained in Step 1. From $\frac{\partial Q}{\partial \mathbf{v}_i} = \mathbf{0}$, $i = 1, 2, \cdots, N$, we have

$$\frac{\partial Q}{\partial \mathbf{v}_i} = 2(\mathbf{v}_i - \mathbf{v}'_i) + 2\beta \sum_{j \in \mathcal{F}_i} \frac{(\mathbf{v}_i^T \mathbf{f}_j - d_j)\mathbf{f}_j}{\|\mathbf{f}_j\|^2} = \mathbf{0}, \quad (14)$$

where $\mathcal{F}_i$ is the set of the faces containing vertex i.

The closed form solution to (14) results in the following equation for computing $\mathbf{v}_i^{n+1}$:

$$\mathbf{v}_i^{n+1} = (\mathbf{R}^{n+1})^{-1} \hat{\mathbf{v}}_i^{n+1}, \quad i = 1, 2, \cdots, N, \quad (15)$$

where

$$\mathbf{R}^{n+1} = \mathbf{I} + \beta \sum_{j \in \mathcal{F}_i} \frac{\mathbf{f}_j^{n+1} \mathbf{f}_j^{n+1\,T}}{\|\mathbf{f}_j^{n+1}\|^2}, \quad (16)$$

$$\hat{\mathbf{v}}_i^{n+1} = \mathbf{v}'_i + \beta \sum_{j \in \mathcal{F}_i} \frac{d_j^{n+1} \mathbf{f}_j^{n+1}}{\|\mathbf{f}_j^{n+1}\|^2}. \quad (17)$$

Our experiments have shown that the algorithm is effective and converges quickly within several iterations.

5.3. Smooth Curved Surface Generation

After a curved stroke is finished, we use a Bezier curve to approximate it to obtain a smooth curve. After we have a (partial) wireframe with identified faces (circuits) and curves represented by Bezier curves, we fill in the circuits denoting curved faces with smooth surface patches. Bilinearly blended Coons patches [6] are used to do it.

6. Experiments

In this section, we show a number of examples to demonstrate the performance of our system. Our system is implemented using Visual C++, running on a PC with 3.4 GHz Pentium IV CPU. The parameter β in (8) is chosen to be 10. Our experiments show that the system is insensitive to the parameters; very similar results are obtained when β changes in $[5, 20]$. The wand tracking and display module work in real time at a rate of 10 frames per second. The system can track the moving of the wand at a maximum speed of about one meter per second. A new user usually needs to take two or three hours of training to get adapted to simultaneous keyboard and wand operation in the system. In our system, strokes are preprocessed and the displayed wireframes are composed of straight lines and smooth curves. The jittering of the strokes by natural hand tremor is smoothed.

Fig. 8 shows a set of wireframes created with our system. For each wireframe in the first column, we give the corrected wireframes in the second column and the 3D reconstruction result displayed in two views in the third and fourth columns. The results show that the correction step (Section 6.2) is effective and the faces of the reconstructed objects are planar.

Figs. 9 shows a set of more complex objects that include strokes of both straight-lines and curves. We can see that our system can handle complex wireframes sketched in the air and generate expected 3D objects.

The time used for the face identification and the generation of planar and curved surfaces of a scene is between 3 and 19 seconds, depending on the complexity of the scene.

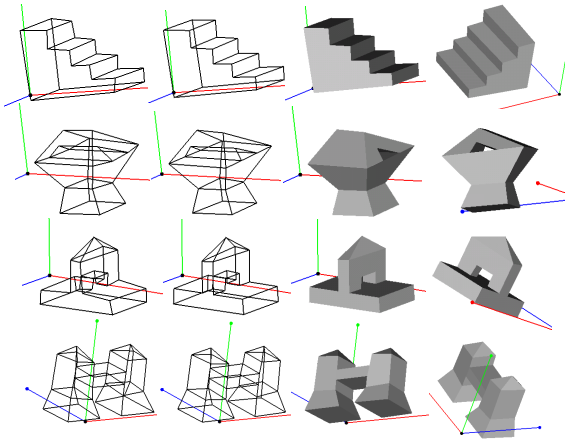

Fig. 8. Experimental results. The axes of the 3D coordinate system are also shown.

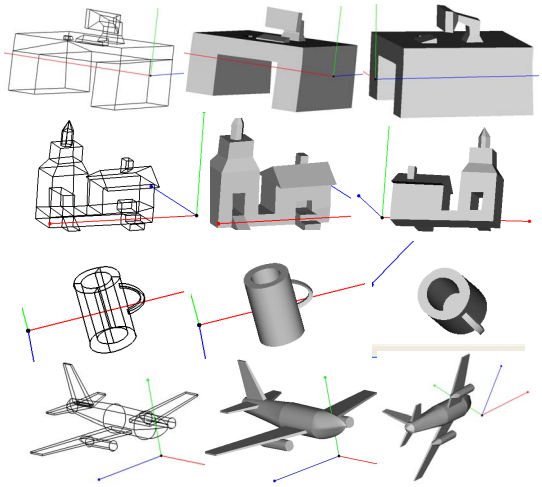

Fig. 9. More experimental results.

For instance, the system takes 3 and 19 seconds to reconstruct the second object in Fig. 8 and the fourth object in Fig. 9 respectively, after the wireframes are obtained.

7. Conclusion and Future Work

We have developed a novel 3D vision-based sketching system with a simple and inexpensive interface allowing users to sketch objects directly in the 3D space. A number of new techniques are proposed for working in this system, including input of object wireframes, gestures for editing and drawing objects, and optimization-based planar and curved surface generation. Experiments have verified its efficacy in designing 3D objects. Our system is still being improved. At the current stage, a new user usually needs to take two or three hours of training to get adapted to simultaneous keyboard and wand operation in the system. Our future work includes: 1) developing more gestures and operations to handle complex objects; 2) improving the tracking algorithm to support more accurate 3D positioning of the wand movement.

8. Acknowledgements

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK 414306).

References

[1] Autodesk Inc. *AutoCAD*. http://www.autodesk.com/.

[2] SolidWorks Corporation. *SolidWorks*. http://www.solidworks.com/.

[3] A. Alexe, L. Barthe, M. Cani, and V. Gaildrat. Shape modeling by sketching using convolution surfaces. *Proc. Pacific Graphics*, 2005.

[4] R. Amicis, F. Bruno, A. Stork, and M. Luchi. The eraser pen: a new interaction paradigm for curve sketching in 3D. *Proc. 7th Intl Design Conference*, 1:465–470, 2002.

[5] Y. Chen, J. Liu, and X. Tang. A divide-and-conquer approach to 3D object reconstruction from line drawings. *ICCV*, 2007.

[6] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, 1997.

[7] T. Grossman, R. Balakrishnan, and K. Singh. An interface for creating and manipulating curves using a high degree-of-freedom curve input device. *SIGCHI Conference*, pages 185–192, 2003.

[8] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. *SIGGRAPH*, pages 406–416, 1999.

[9] O. Karpenko and J. Hughes. Smoothsketch: 3D free-form shapes from complex sketches. *SIGGRAPH*, pages 589–598, 2006.

[10] D. Keefe, D. Feliz, T. Moscovich, D. Laidlaw, and J. LaViola. Cavepainting: a fully immersive 3D artistic medium and interactive experience. *Symposium on Interactive 3D Graphics*, pages 85–93, 2001.

[11] H. Lipson and M. Shpitalni. Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer-Aided Design*, 28(8):651–663, 1996.

[12] J. Liu, L. Cao, Z. Li, and X. Tang. Plane-based optimization for 3D object reconstruction from single line drawings. *IEEE Trans. PAMI*, 30(2):315–327, 2008.

[13] J. Liu and Y. Lee. A graph-based method for face identification from a single 2D line drawing. *IEEE Trans. PAMI*, 23(10):1106–1119, 2001.

[14] J. Liu, Y. Lee, and W. K. Cham. Identifying faces in a 2D line drawing representing a manifold object. *IEEE Trans. PAMI*, 24(12):1579–1593, 2002.

[15] J. Liu and X. Tang. Evolutionary search for faces from line drawings. *IEEE Trans. PAMI*, 27(6):861–872, 2005.

[16] S. Schkolne, M. Pruett, and P. Schroder. Surface drawing: creating organic 3D shapes with the hand and tangible tools. *SIGCHI Conference*, pages 261–268, 2001.

[17] A. Shesh and B. Chen. Smartpaper: An interactive and user friendly sketching system. *Proc. Eurograph*, 2004.

[18] K. Sugihara. *Machine Interpretation of Line Drawings*. MIT Press, 1986.

[19] R. Zeleznik, K. Herndon, and J. Hughes. Sketch: an interface for sketching 3D scenes. *SIGGRAPH*, pages 163–170, 1996.