



Highly interactive 3D geometric modeling

J. Liang, M. Green

*Department of Computing Science, University of
Alberta, Edmonton, Alberta, Canada, T6G 2H1*

ABSTRACT

This paper presents an effort to build a 3D modeling system that can be used for 3D quick prototyping. We focus on providing a highly interactive user interface for 3D direct manipulation, and allowing for the efficient creation and modification of 3D shapes. The main features of the system are (1) the use of kinetic depth effect to provide effective 3D visualization on a traditional CRT screen, (2) the use of a hand-held six degrees of freedom input device (a bat) for direct manipulation of 3D objects, and (3) a set of modeling operations based on the new input device. The system allows a designer to quickly create, modify, place, orient, and organize 3D objects in a natural and intuitive manner. Preliminary results show that such a highly interactive interface has the potential of increasing the efficiency of 3D geometric modeling by a large factor.

1 INTRODUCTION

Three dimensional geometric modeling has been a challenging topic for computer graphics and computer-aided design [11]. Although many modeling systems have been developed to facilitate 3D geometric modeling, few have penetrated into the early stage of the design process, that is, the conceptualization or sketching phase [4]. The major reason for this is the generally low efficiency of interaction. Using a commercial system, it usually takes several hours to model a mechanical part of moderate complexity [6]. This amount of turn-around time makes it virtually impossible to “sketch” with a computer [5].

We believe that this difficulty in quick prototyping is largely due to poor user interface design. In particular, the traditional mouse-based interaction style is a bottleneck to 3D geometric design, as it forces the designer to decompose a 3D task into a series of 1D or 2D tasks.

Based on this observation, we have designed a new highly interactive user interface for 3D geometric design. In our system, the designer is equipped with high degree-of-freedom input devices. They allow the designer to directly manipulate objects in 3D space, without having to map the task into 1D or 2D space. The naturalness and intuitiveness of such interactions can greatly enhance the efficiency of geometric design, thus making it possible to sketch with a computer. Using our system, we show that the prototyping time can be reduced to about one tenth of that required by commercial systems.

In the following section, we discuss the problems with mouse-based interaction techniques and examine some experiments on using high degree-of-freedom input devices for modeling tasks. In section 3, we discuss the overall setting of our system, and several basic design issues. In section 4, we discuss the use of the bat in modeling operations. Some preliminary results are presented in section 5.

2 PREVIOUS WORK

Mouse-based 3D manipulations have two major limitations. First, simultaneous translations along all three directions are not possible, due to the 2D nature of the mouse. This is the fundamental obstacle barring mice from being useful in 3D quick prototyping. When a designer wants to create a box-shaped object, he/she wishes to do it in a single stroke, as in the case of creating a rectangle with a 2D drawing tool. If a system forces the designer to break such a task into sub-tasks (such as creating a rectangle, and then adding the third dimension), the designer will have to think in terms of subspaces, rather than focusing on the 3D nature of the object. In other words, the level of abstraction will be lowered, and the result is a low interaction efficiency.

Second, 3D rotations are not effectively supported. Most techniques limit the rotations to around one of the major axes at a given time. To effect a general 3D rotation, three consecutive rotations have to be performed. The fact that rotations are not commutative makes the situation worse. As a result, it is difficult to orient a 3D object in space.

To overcome these problems, several experiments have been carried out in using high degree-of-freedom interaction techniques to support 3D modeling. Of them, three are of great importance.

In mid 1970's, James H. Clark built an experimental system for designing surfaces in 3D [2]. It used a head-mounted display to provide a

stereoscopic view to the designer, and let the designer use a mechanical 3D “wand” to drag control points in 3D. The system offered the designer a strong kinaesthetic correspondence which gave it a great advantage over the more common 2D techniques of viewing and designing curves and surfaces.

Emanuel Sachs et al at MIT built the 3-Draw system for interactive 3D shape design [9]. The system was based on a pair of Polhemus Isotrak six degree-of-freedom tracking devices. The designer held one sensor (in the form of a palette) in his left hand to specify a moving reference frame, and used another sensor (in the form of a stylus) to draw and edit 3D curves in space. A graphics display was used to visualize the scene from a virtual camera position.

Butterworth et al built the 3DM system at University of North Carolina [1]. The system uses a head-mounted display to place the designer in the modeling space. The input device is a Polhemus 3Space Isotrak mounted in a hollowed-out billiard ball with two buttons on it. The designer used the hand-held tracker to select functions from a toolbox, and to create/manipulate objects in 3D space.

Both 3-Draw and 3DM offered easy and intuitive ways of modeling 3D shapes, and demonstrated that 3D direct manipulation interfaces are superior to mouse-based ones. However, the lack of ergonomic considerations and support for precision in modeling made them unusable for engineering purposes.

3 SYSTEM OVERVIEW

Our system is based on an IRIS 4D/310 VGX graphics workstation and a pair of Polhemus Isotrak six degree-of-freedom tracking devices. We use one tracking device to dynamically monitor the designer's head position, and provide the kinetic 3D display effect [10] [8]. We use another six degrees of freedom tracker as a hand-held bat [12]. It is the main input device for our system.

3.1 3D Display and Input

To enable 3D direct manipulation, a system must first provide enough spatial information to the designer. Since we don't have true 3D display techniques yet, we have to rely on 2D displays to visualize the 3D scene.

In addition to common depth cues such as perspective projection, hidden surface removal, and shading, we also supply kinetic depth effect by tracking the designer's head position. When sitting in front of the graphics workstation, the designer wears the sensor of the tracking device on his head. Through a series of transformations, the approximate eye position with respect to the display window is obtained and used as the

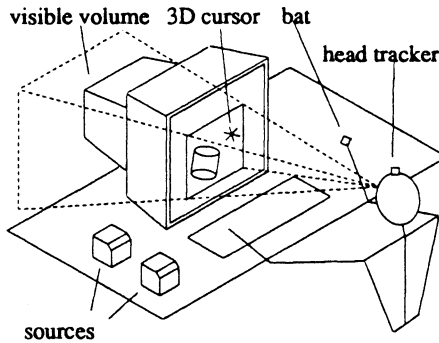


Figure 1: A typical setting of our system.

center of the perspective projection. Since the display is updated at 15 to 60 frames per second, the designer has a strong spatial perception as if he/she is looking into a real 3D space. Similar experiments with kinetic depth effect were also made by other researchers [10] [3].

This setting provides a platform for 3D direct manipulation.

The main input device for our system is a bat — a hand held Isotrak sensor. The position and orientation of the bat is monitored in real time, and is echoed on the display as a 3D cursor (in the form of a jack). The designer uses the bat to interact with objects and to navigate through the design space.

Many design issues have to be carefully treated before one can make good use of such an environment.

3.2 Coordinate Transformation

In order to achieve correct kinaesthetic correspondence between hand movement and the 3D cursor motion, the tracking data from the bat has to go through a series of transformations.

First, the measurement from the Isotrak is usually noisy and bears a certain lag [7]. So we need to filter it first.

Second, since the tracking data are given in the device coordinate system D , they have to be transformed into a useful reference frame T , in our case, that of the graphics terminal.

Third, since the Isotrak is an absolute locator, a clutch mechanism is necessary for re-mapping its physical location into a logical location. In addition we found two kinds of relocation operations to be very useful: (a) a “homing” operation which brings the 3D cursor to the center of the visible volume, (b) a “masking” operation that disables the passing of either positional or orientational change from the bat to the user interface. With these mechanisms, the bat can be operated with the same ease as a relative locator (such as a mouse).

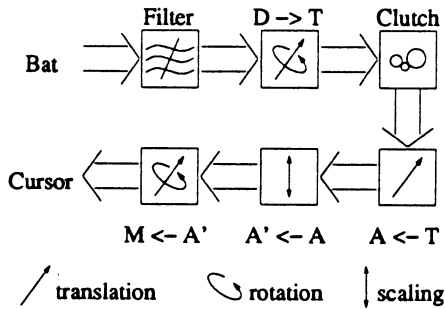


Figure 2: Transformations from bat to 3D cursor.

Fourth, certain 3D manipulations require a reference frame other than the terminal or the modeling coordinate system. For example, to inspect an object and its relationship with the neighboring objects, one may grab it with the bat and simultaneously rotate/translate it. Such operations require a reference frame whose origin is near the center of the visible volume. Zooming operations also have a similar requirement. For this purposes, we define an *apparent coordinate system* A , which is centered at the visible volume. The translation from T to A is dynamically determined by the viewpoint (and also by the window size).

Fifth, any zoom factor should be applied to the positional part of the bat data before it is transformed into the modeling coordinate system M . This ensures a constant ratio between hand movement and the apparent cursor motion. So, at a zoom factor of 10, the 3D cursor still seems to move 1 cm to the designer when the hand moves 1 cm, although the distance it travels in the modeling coordinate system M is actually 0.1 cm.

3.3 Navigation in Space

Since the designer has a space of virtually unlimited volume to work with, but at any given moment can only see a part of it (i.e. the visible frustum), some forms of navigation are necessary.

We found the following four types of navigation useful.

First, a “grabbing” operation lets the designer move the entire model with the bat. This mechanism, when used in conjunction with zooming, provides a natural way of bringing any part of a model into the visible volume.

Second, a “focus” operation brings a selected object to the center of the apparent coordinate system, i.e. the visual sweet spot, so subsequent manipulations on the object become convenient. In order to maintain visual continuity, we use an animation sequence to provide a smooth transition. The system first determines the minimum translation and/or rotation needed for the focus operation, then divides them into 10 parts and

carries them out in about half second (the update rates of the system is about 20 frames per second).

Third, a “fly-with-bat” operation lets the designer move around in space, and inspect different parts of the model. This is useful for tasks such as architectural walk through. Since the apparent flying speed is zoom-invariant, one can effectively change the pace by adjusting the zoom factor.

Fourth, the designer can mark a particular view setting by putting down a “spacemark”. Later, he/she can come back to that setting by traversing the chain of spacemarks. Also, the designer can always bring the model to its canonical positional by a “homing” operation.

These navigation mechanisms are very useful when the model is large and complex.

4 MODELING OPERATIONS

The most important advantage of a bat over a mouse is the ability to create and make changes to objects by 3D direct manipulations. The fact that the designer is in the same space as the geometric model relieves the designer from having to break down 3D tasks into 2D or 1D operations, and thus makes the interface “transparent”. We will discuss the design of basic modeling operations in the following sections.

4.1 Creation by Rubber-Banding

A primitive solid can be created in a single stroke using 3D rubber-banding. The designer first chooses a shape with the bat using a “daisy” menu [8], then presses a key to set the current position of the 3D cursor as one corner of the bounding box, the other corner of the bounding box follows the 3D cursor as it moves with the bat. When the solid is stretched to the right shape, the designer releases the key to finalize the solid.

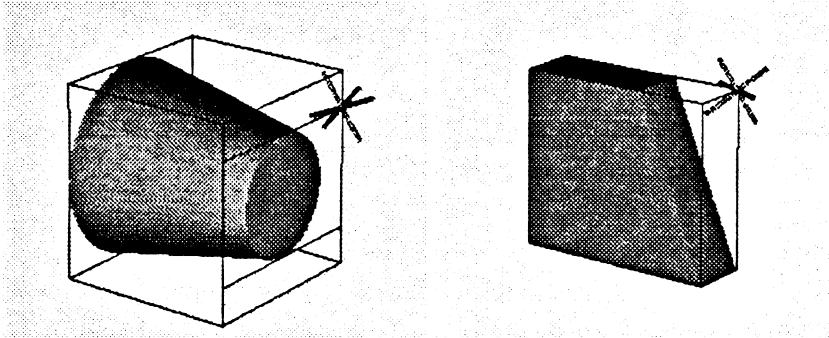
For shapes with three axial measurements (such as a box), the mapping from the bounding box to object geometry is trivial. However, certain shapes have non-axial measurements (such as a circular shape) or more than three measurements (such as an extruded trapezoid). In those situations, proper mappings have to be devised.

Initially, we tried to employ the data of all six degrees of freedom simultaneously. It proved to be the wrong choice. We observed three facts. First, controlling all six degrees of freedom simultaneously tends to overload the hand. Second, orientation appears intrinsically more difficult to control than position. Third, mappings from an orientation change to a position change are usually difficult to understand and use.

Based on these observations, we adopted the following two rules.

First, if the shape in question has three measurements, but some of them are not axial in nature, we associate each one with an axial change.

For example, a truncated cone has three measurements: the height, the diameter of the top, and the diameter of the base. We map the width of the bounding box to the top diameter, and the length of the bounding box with the base diameter. (The height of the bounding box obviously maps to the height of the truncated cone.) In this way, we made good use of the ability of the designer to control 3D positions.



(a) A truncated cone.

(b) An extruded trapezoid.

Figure 3: Rubber-banding creation of primitive objects.

Second, if the shape in question has more than three measurements, we only directly control three of them, and compute the rest with a predefined proportion. In this way, the designer can quickly get the object in roughly the desired shape and then come back to fine tune it. For example, an extruded trapezoid has four measurements: the width, the height, the bottom length, and the top length. We map the length of the bounding box to the bottom length, and keep the top length at a fixed proportion to it. We use the golden proportion as a default for these cases. The linear mapping makes it equally easy to set either the top length or the bottom length to a desired amount.

4.2 Region-Based Reshaping

Modification to a model is at least as important as its creation, and usually occurs much more frequently than creation. We designed a set of reshaping operations that take advantage of the bat.

In an ordinary 2D drawing system, the reshaping operation is often based on a set of “handles” on the object. The user drags a handle with the mouse to enact a certain type of reshaping. Although this method can be directly extended into 3D, we found it less effective than in the



2D case. The measurement noise and lack of sufficient depth cues often makes it difficult to place the 3D cursor at an exact handle point.

We designed a region-based method to facilitate the reshaping operation. The idea is to extend the handles into 3D regions, so the precision requirement can be lowered. The space around the object is divided into different regions, each of which corresponds to a different type of reshaping operation. After the designer selects an object, he/she moves the 3D cursor into one of the regions before commencing the reshaping action, to specify the type of the reshaping action. As a visual feedback, the 3D cursor takes different shapes in different regions. When the designer presses a key to start reshaping, the system attaches certain measurements of the object to the the 3D cursor. Releasing of the key finalizes the operation.

The following figures illustrate the region-based reshaping of a cylinder. Figure 4 (a) shows that the 3D cursor takes the shape of a two-ended arrow when it is near the center of the cylinder end, meaning that should the designer start to reshape, he/she will be changing the length of the cylinder. Figure 4 (b) shows that the 3D cursor changes its shape to a bow-arrow when it is near the middle of the cylinder, meaning that a reshape operation will change the radius of the cylinder. Figure 4 (c) shows that the 3D cursor has both shapes on it when it is near the "rim" of the cylinder, meaning that both the height and radius of the cylinder may be changed simultaneously. Figure 4 (d) shows a reshaping operation changing both the height and radius of the cylinder. At the beginning of a reshaping operation, the 3D cursor is first snapped to the corresponding point on the object, so the object will not undergo a sudden change in shape.

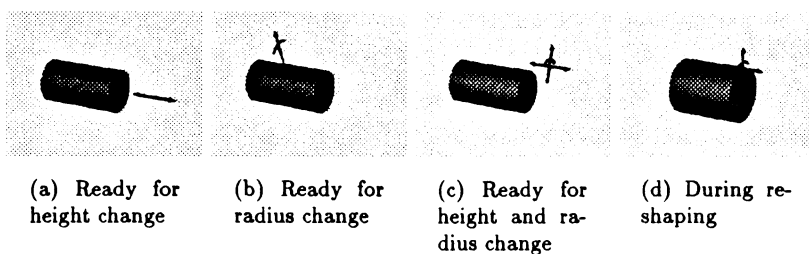


Figure 4: Region-based reshaping.

Composite objects can be reshaped via their bounding boxes, in one, two, or three dimensions. If the starting position of the cursor is near the center of a face, then the object will be stretched only along the dimension

perpendicular to that face. If the cursor started near the center of an edge, then only the two dimensions perpendicular to that edge will change with the cursor. Similarly, the designer can pick a corner and change all three dimensions at the same time.

4.3 Placement, Alignment, and Others Operations

Translations and rotations of an object with the bat are straightforward. The designer first selects the object, then presses the key for translation, rotation, or both, and the object moves with the 3D cursor as the bat moves. Releasing the key finalizes the position and orientation of the object.

To defeat the measurement noise of the bat and the unsteadiness of the hand, an optional snap grid can be turned on for position and/or orientation.

The bat is also used to specify an axis for certain operations. Since the shape of the bat makes its X-axis very distinct from the other two, we use it as the direction of the bat. The designer points the bat at roughly the direction of the desired axis, and presses a key to instruct the system to find the nearest axis to that direction. The system echoes the axis it chose as a thin rod, and the designer can start a constrained translation, rotation, reshaping, or alignment between selected objects, with respect to that axis. This natural way of specifying an axis eliminates the need for the designer to know the names of the axes.

Once an axis is specified, the designer may also perform 'conformation operations' to change the size of an object according to some reference objects. Such operations relieve the designer of repeatedly entering the common dimensions in a design.

For visual continuity, all the changes in position, orientation, and size are visualized with animation. Each command results in a 10-frame animation sequence, and is carried out in about half a second. This eliminates the "jumping" of objects, which often require the user to mentally re-assimilate the scene.

The designer may also select several objects and group them together or perform CSG operations. A hierarchical model can be built in this way.

5 CONCLUSIONS

In this paper, we proposed a new user interface for interactive 3D geometric modeling. The aim is to remove the bottleneck between the designer and the modeling system by using high degree-of-freedom input devices and novel interaction techniques. We combined a kinetic 3D display scheme with hand held bat input to provide the designer with a 3D direct manipulation interface. We discussed coordinate transformations

for achieving the correct kinaesthetic correspondence, which is vital in such an environment.

Although only at a preliminary stage, the system has shown to be very efficient in 3D quick prototyping. Compared with commercially available systems, it usually takes only one-fifth to one-tenth of the time to build a prototype. For example, the simple mechanical assembly in figure 5 can be modeled in about half an hour.

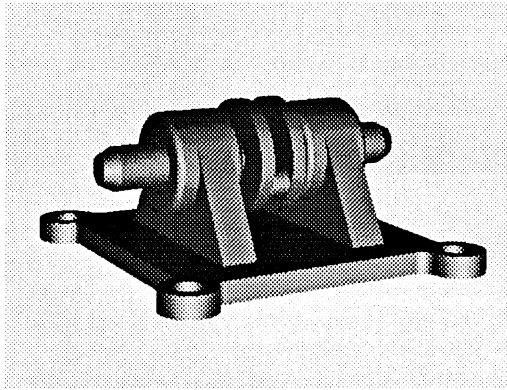


Figure 5: A simple assembly modeled in about half an hour.

Our next steps are to incorporate geometric constraints in this modeling environment, to further increase the efficiency in modifying 3D models, and to experiment with free-form surface design using the bat.

ACKNOWLEDGEMENTS

The first author wishes to thank Chris Shaw and Dani Beaubien for many discussions and comments.

REFERENCES

1. Butterworth, J., A. Davidson, S. Hench, and T. M. Olano, '3DM: A Three Dimensional Modeler Using a Head-Mounted Display,' *Proc. 1992 Symposium on Interactive 3D Graphics*, Cambridge, Massachusetts, March 29–April 1 1992, pp. 135–138.
2. Clark, James H., 'Designing Surfaces in 3-D,' *Communications of the ACM*, Vol.19, pp. 454–460, 1976.
3. Deering, Michael, 'High Resolution Virtual Reality,' *Proc. ACM SIGGRAPH '92*, Chicago, Illinois, July 26–31, 1992, pp. 195–202.
4. Encarnação, J., 'CAD Technology: A Survey on CAD Systems and Applications,' *Computers in Industry*, Vol.8, 145–150, 1987.

5. Hatvany, Józef, 'Can Computers Compete with Used Envelopes?' *Proc. IFAC 9th Triennial World Congress*, Budapest, Hungary, 1984, pp. 3373-3375.
6. Hodskinson, M., and R. Patel, '3-D Modeling in a Teaching Company Scheme,' *Computer-Aided Engineering Journal*, Vol.9, pp. 177-180, 1989.
7. Liang, J., C. Shaw, and M. Green, 'On Temporal-Spatial Realism in the Virtual Reality Environment,' *Proc. ACM Symp. on User Interface Software and Technology*, Hilton Head, South Carolina, November 11-13, 1991, pp. 19-25.
8. Liang, J., and M. Green, 'Quick Prototyping for Engineering Design,' to appear in *Proc. ASME Symposium on Computers in Engineering*, Houston, Texas, January 31 - February 4, 1993.
9. Sachs, Emanuel, Andrew Robers, and David Stoops, '3-Draw: A Tool for Designing 3D Shapes,' *IEEE Computer Graphics and Applications*, Vol.11, pp. 18-24, 1991.
10. Schmandt, Christopher, 'Spatial Input/Output Correspondence in a Stereoscopic Computer Graphics Work Station,' *Computer Graphics*, Vol.17, pp. 253-261, 1983.
11. Sproull, R. F., 'Parts of the Frontier Are Hard to Move,' *Computer Graphics*, Vol.24, p.9, 1990.
12. Ware Colin, and Danny R. Jessome, 'Using the Bat: A Six Dimensional Mouse for Object Placement,' *Proc. Graphics Interface '88*, Edmonton, Alberta, June 6-10, 1988, pp. 119-124.