

Interactive 3D Geometrical Modelers for Virtual Reality and Design

Mark Green*, Jiandong Liang**, and Chris Shaw*

*Department of Computing Science, University of Alberta, Edmonton, Canada

**Alberta Research Council, 6815 8 St. N.E., Calgary, Alberta, Canada

Abstract

Most geometrical modelers for 3D objects use 2D interaction techniques and make little use of 3D display techniques. This interaction style forces the user to decompose 3D design operations into sequences of 2D interface operations, and there is an extra cognitive load of interpreting the 2D displays as a 3D object. While this may be acceptable for drafting applications, it has a significant negative effects on design applications.

The recent interest in 3D graphics and Virtual Reality suggests that 3D interaction techniques and displays will provide a better environment for 3D geometrical design. This paper explores this idea by briefly discussing the requirements for geometrical modelers for VR and design, presenting some of the new 3D interaction techniques that we have developed, and briefly describing some of our geometrical modelers.

1. Introduction

There is a rich literature in geometrical modeling, and many 3D geometrical modelers have been constructed, both as research projects and as commercial products. Virtually all 3D geometrical modelers restrict the user to 1D and 2D interaction techniques, and rarely use 3D display techniques to show the object under design. We argue that 3D geometrical design should be performed in a 3D environment. With conventional geometrical modelers, the designer must decompose 3D modeling operations into sequences of lower dimensional operations, shifting his or her concentration from the design task to the task of interfacing with the geometrical modeler. Similarly, the use of 2D displays forces the designer to mentally assemble several 2D representations to produce the object's true 3D shape. Both of these factors increase the designer's cognitive load and interfere with the design process. Thus, designer productivity could be increased by providing geometrical modelers that support 3D interaction and display techniques.

The addition of 3D operations to geometrical modelers must be carefully considered. While 3D operations are obviously important in 3D design, there are still a large number of operations that are basically 1D or 2D. Our preliminary studies have shown that 3D operations are mainly used in three types of design tasks. The first type is rough sketching operations that occur in the early stages of design. At this point the designer is mainly interested in the overall shape of the object, and precision is not a consideration. The second design task is establishing a coordinate frame for subsequent operations. These operations are usually 1D or 2D in nature, but they must be performed when the object is in a particular orientation, which can best be achieved using 3D operations. The third design task is shape examination, where the designer examines either the overall or detailed shape of the object. A 3D display of the object, where the object's position and orientation can be freely modified, is an efficient way of accomplishing this task. These studies have also indicated that designers prefer 1D and 2D operations for detailed design and where precision is required. The lower dimensionality gives the designer more control over the one or two aspects of the shape that are modified in these tasks.

There is a temptation to produce geometrical modelers that are restricted to 3D interaction and display techniques, and a number of well known research prototypes have used this approach [1, 5]. In light of the above observations, this approach seems to be as bad as restricting the geometrical modeler to 1D and 2D operations. A more profitable approach is to integrate 3D operations with the existing 1D and 2D operations that are used in existing geometrical modelers. This paper summarizes some of the work that we have done along this line. The next section of this paper presents the requirements for geometrical modelers that we have developed. The third section briefly describes some of the 3D interaction techniques that we have developed for the basic tasks in geometrical modeling. The fourth section presents some of the geometrical modelers that we have produced over the past four years, and shows how they reflect the basic theme of this paper.

2. Requirements for Geometrical Modelers

Over the years we have developed a set of requirements for geometrical modelers. These requirements are biased by our particular application domains, and may not reflect the needs of other domains. The first application domain is the production of 3D models for virtual environments. This was the motivation for our original research in 3D geometrical modeling. The second application domain is Industrial Design, which is the result of several joint research projects with our Art and Design Department. These application domains tend to produce a slightly different set of requirements than areas addressed by most geometrical modelers, such as drafting or mechanical design. In our application areas there is more emphasis on exploring the design space, and less emphasis on precision.

The main requirements for geometrical modelers in our application domains are:

- 1) **Successfully Compete With the Best Geometrical Modeling Tools.**
Industrial designers work with a wide range of modeling tools, the majority of which are not computer based. Computer based design tools should compete with pencil sketches and clay models. While computer based techniques have some advantages over these other media, such as a digital representation of the resulting design, they tend to suffer in the areas of time and visual representation. It should be possible to produce a rough sketch of a design using a geometrical modeler in about the same amount of time as a designer using pencil and paper. Similarly, display techniques should be capable of giving the same impression of the object's shape as a physical model. This doesn't imply photo-realistic rendering, but the ability to pick up the object and examine it from any direction.
- 2) **Access to the Complete Range of Modeling Tools.**
A wide range of mathematical and algorithmic techniques have been developed for geometrical modeling, but only a fraction of these techniques are widely available in interactive geometrical modelers. High quality interaction techniques for state of the art modeling techniques must be available in interactive geometrical modeler, so the user can take advantage of the power of these techniques.
- 3) **Ease of Learning and Use by Designers.**
In order to be successful it must be possible for a designer to become fluent in the use of geometrical modeler in a short period of time. This can be quite a challenge, since most modelers have a rich function set. This implies that the modeler should be structured in such a way that the designer can be productive once a small set of features have been mastered, and learning new features is a relatively simple task.
- 4) **Exploration of the Design Space.**

A geometrical modeler must support the rapid exploration of the design space. In the early phases of the design process, the designer only has a rough idea of the object's shape, and may produce many designs before reaching the final product. It must be possible to very quickly sketch out a design, and then modify the design in a way that matches the designer's thought process. The production of a single design shouldn't be so time consuming that the designer is discouraged from exploring the design space.

In addition to the above requirements, there is one further requirement for geometrical modelers that are used for virtual reality. In this application domain the designer must be able to control the geometrical complexity of the resulting object. Loosely speaking, the geometrical complexity is the number of geometrical primitives that are required to represent the object, and thus governs its display time. There are two aspects to this requirement. First, the designer must be able to budget geometrical complexity based on the display time available for each object. That is, given a fixed amount of time for display, the objects must be designed appropriately, and the geometrical modeler should assist in this budgeting process. Second, the geometrical modeler should be able to produce multiple representation of the same object with different levels of geometrical complexity. This allows the virtual environment to switch the object's representation based on the amount of display time that can be allocated to it.

3. Interaction Techniques

The aim of this work is to add 3D interaction and display techniques to geometrical modelers. Existing modelers support a wide range of 1D and 2D interaction techniques, but force the user to decompose 3D tasks into a series of operations supported by these interaction techniques. With the addition of 3D interaction techniques, a common space that supports all the interaction techniques must be selected. Our work is based on the use of a 3D space supported by 3D input devices and display techniques. This approach has the following two advantages. First, performing all the operations with the same 3D input device, reduces the amount of device switching. If 3D interaction is simply tacked onto an existing geometrical modeler, then the user must switch between the standard 2D device and the 3D input device, whenever a 3D operation is performed. Second, using a common 3D display space for all operations reduces the user's cognitive load. If 2D operations are performed in a 2D display space, or an orthographics projection of the 3D space, then the user must re-assimilate the 3D object after each 2D operation.



Figure 1 Bat input device

There are a number of interaction techniques that are used in geometrical modelers, and at least the following four interaction tasks must be supported. First, the user must be able to select commands to be performed. The most common way of doing this is through menu selection. Other

techniques, such as gesture recognition, have been suggested for this task, but to date menus are the most effective technique. Second, the user must be able to select objects and sub-objects in the environment. Third, the user must be able to modify the shape of a selected object. Fourth, the user must be able to examine objects. The user must be able to view the external shape of the object, as well as its internal structure. Interaction techniques for performing these four tasks are described in the following sections.

All the interaction techniques described in this section make use of the Bat [7] as their input device. A Bat is a 3D tracker that reports 3D position and orientation data. In addition, three buttons are mounted on the bat as a way of signaling events. Figure 1 shows the Bat that we use with our geometrical modelers.

3.1. Menu Selection

The obvious approach to menu selection in 3D is to embed a 2D menu in 3D space. This can easily be done by drawing a standard 2D menu on a flat surface in 3D, and then pointing at the command to be selected. We have tried this approach, and found that it didn't work, since the flat menu presentation interrupted the user's perception of the 3D space. As a result, we have developed several menus that have a 3D shape and can be placed in the 3D space without disrupting the perception of it.



Figure 2 Daisy (left) and ring (right) menus

The first 3D menu we developed is called the daisy menu and it is based on distributing the menu items over the surface of a sphere. The center of the sphere is attached to the Bat, and the menu can be moved and oriented by performing the same operations on the Bat. Menu selection is based on the use of a selection cone. The cone's apex is placed at the center of the sphere, and its base points towards the user. The size of the cone is designed in such a way that it can contain only one menu item. The daisy menu is usually used as a pop up menu. When the user presses a Bat button the menu appears. With the button depressed, the user rotates the menu until the desired item is in the selection cone, and then releases the button.

The daisy menu has been used in one of the geometrical modelers that is described in section 4. This menu is better than using a 2D menu, but it does have two problems. First, with most 3D trackers, some rotations are easier to perform than others. Any tracker with a cable attached to it has a natural rotation axis that is defined by the cable. Rotating about this axis is easier than rotating about other axis. Thus, some items on the menu are easier to select than other items. Second, when the menu is static, it can be quite difficult to determine which item is at the front of the menu, and which is at the back. This can cause some selection errors.

The ring menu was developed to solve some of these problems. In this type of menu the menu items are distributed around a ring, and this ring is perpendicular to the natural axis of the tracker. The menu items are enclosed in two semi-transparent concentric rings, with the outer ring having a gap that forms the selection region. To select an item, the user rotates the Bat about its natural axis until the desired item is in the selection region, and then indicates the selection. Since the rotation is about the tracker's natural axis, all items can be selected with equal ease. The concentric rings make it easy to determine the item that is at the front of the menu. The ring menu is used in two of the geometrical modelers described in section 4.

3.2. Object Selection

One of the major problems in object selection is depth complexity. Each pixel can be covered by several objects, with each object being a different distance from the user. Normally only the closest object to the user is displayed, but this may not be the object that the user wants to select. Thus, interaction techniques that are based on selecting a point on the display surface, and selecting the object that is displayed at that point, won't work all of the time. This raises the question of why the user would want to select an object that isn't visible? An object that isn't the closest the user at a particular pixel may still be visible to the user for at least two reasons. First, the closest object could be transparent, allowing the user to see the objects that are behind it. Second, a small part of the object may be visible in another part of the screen, but it may not be convenient for the user to select the object at that place.



Figure 3 Use of the spotlight selection technique

One approach that has been used to solve this problem is to shoot a ray from the Bat in the direction that it is pointing, and then select the first object intersected by this ray. This technique doesn't work particularly well for small or distant objects. In this case of small objects, it can be quite difficult to point the ray at the object, and in the case of distant objects, a small shift in the angle at the Bat causes the ray to move a considerable distance when it is far from the Bat.

The solution to these problems is to use the spotlight selection technique. In this interaction technique the Bat is used to control the orientation of an infinite cone with its apex located at the tracker position. In the simplest version of this technique, the axis of the cone is aligned with the natural axis of the Bat, and the user selects objects by aiming the Bat at them. Since the selection cone has a finite area that increases with distance from the Bat, it is quite easy to select small and distant objects. With this interaction technique feedback can be provided by using a spotlight light source located at the tracker and point along the cone's axis.

This interaction technique can be optimized in the following way. Instead of pointing the cone's axis along the Bat's natural axis, a line from the user's position to the Bat is used. In this way the object that is directly under the Bat will be selected, if the user clicks the Bat button. This allows the user to quickly select the closest object, and by re-orienting the Bat, select any object in the environment.

3.3. Object Manipulation

A number of interaction techniques are available for 3D object manipulation. Most of the techniques described in this section are not unique to our work, but the way that they have been integrated into our geometrical modelers is.

A form of 3D direct manipulation is used for object creation in two of our geometrical modelers. After the user has selected the object to create, the Bat is moved to one corner of the space occupied by the object and the user presses a button. While the button is depressed, the user can move the Bat to define the volume occupied by the object. While this is happening, the object is displayed within the volume to provide feedback to the user. When the button is released, an instance of the object is created within the volume specified by the user. This interaction technique takes advantage of three of the degrees of freedom provided by the tracker to define the volume occupied by the object. This works well for objects that are defined by three parameters, such as boxes, but for other objects, such as pipes, that have more than 3 parameters, at least one of the parameters must be left unspecified. In our current implementations, these extra parameters are made a function of one of the other parameters. For example, in the case of a pipe, the interior diameter is a function of the exterior diameter. After the object has been created, the extra parameters can be modified using other interaction techniques. Most trackers have more than three degrees of freedom, such as the tracker's orientation, that could be used to control more than 3 shape parameters. In practice we have found that this doesn't work very well, since it is hard to manipulate the tracker's orientation without changing its position.

Most of the shape modification interaction techniques are 2D interaction techniques that are embedded in a 3D space. The edges and faces of the sub-objects used to construct an object define a number of 1D and 2D coordinate systems. These coordinate systems can be used as the basis for a number of manipulations. For example, when one object is placed with respect to another object, the first object defines a number of coordinate systems that constrain the second object's motion. When the second object is touching a face of the first object, it can be constrained to move in the plane of that face, or perpendicular to the plane of that face. This greatly simplifies object positioning for CSG operations, where the accurate positioning of objects is quite important.

3.4. Object Examination

Object examination in 3D is a natural operation. The user selects an object, using the interaction technique described in section 3.2, and once selected it becomes attached to the tracker and moves with it. Thus the user can move and orient the object in any way, allowing the user to view the object from any position or orientation. In addition, the object can be scaled to provide a convenient size for observation.

This interaction technique can be extended in two ways to provide a view of the object's interior structure. First, parts of the object can be made transparent to see how they interact with each other. This is particularly helpful for the difference operator in CSG, since it gives the user a good impression of the final result of this operation. Second, user controlled clipping planes can be

added to remove parts of the object in order to view its internal structure.

4. Geometrical Modelers

This section briefly describes three geometrical modelers that use the interaction techniques presented in the previous section. The first two modelers are solid modelers and are both based on the same basic interaction style, and the third modeler is a free form surface modeler.

The first modeler, called JDCAD+, is a geometrical modeler for VR applications that is based on our first solid modeler, JDCAD [3]. In this modeler the primitive solids are selected from a daisy menu, and the interaction techniques described in section 3.3 are used to create and manipulate objects. Complex objects are constructed from simpler objects using a join operator. This operator creates a modeling hierarchy, with one object as the root of the tree, and the other objects as the children. Each of the nodes in the hierarchy has a transformation matrix that positions to corresponding sub-object within the object.

In addition to geometrical modeling, JDCAD+ also supports a number of operations for behavioral modeling. The user can add transformations to the modeling hierarchy that are functions of time. This can be used to produce periodic motions, such as a rotating wheel or airplane propeller. Keyframing can also be used to produce object behavior. In this approach the user creates a sequence of keyframes by manipulating the object's sub-parts and recording their transformation matrices. When the behavior is triggered at run-time these matrices are interpolated to produce the motion.

The second geometrical modeler, JDCAD 2.0, extends the previous solid modeler by adding CSG operations, and uses a better set of interaction techniques [4]. The ring menu described in section 3.1 is used for primitive object selection in this modeler. The interaction techniques described in section 3.3 can be used to create sub-objects, modify their shape, and position them with respect to each other. In addition, CSG operations can be performed between any two objects in the model. Most of these operations are performed in less than a second, so the interactive feel of the modeler is preserved. The object examination techniques described in section 3.4 are used to inspect objects before the CSG operations are performed.

This modeler has been used in two user studies. In the first study it was compared to AutoCAD. In this study a number of students from an AutoCAD course were selected and asked to create a number of objects from an AutoCAD textbook, and the time required to construct these objects was recorded. The same students were given a short introduction to JDCAD 2.0 and several hours of practice. They were then asked to produce the same models using JDCAD 2.0, and the time required to construct them was recorded. Approximately half the students constructed the model in AutoCAD first, and half in JDCAD 2.0 first, to remove one source of bias in the results. This study showed that JDCAD 2.0 was significantly faster than AutoCAD. A second study was performed using students in Industrial Design as the subjects. In this study the students were asked to sketch a mechanical part using pencil and paper, and the time required to complete the sketch was recorded. An experienced JDCAD 2.0 user constructed the same models using JDCAD 2.0, and the time required to construct the models was recorded. The student's sketching time was longer than the time required for JDCAD. A subsequent informal study with a more experienced designer produced sketching times that were the same or slightly faster than JDCAD 2.0. This shows that it's possible to construct geometrical modelers that can compete with paper and pencil in terms of sketching time.

The third geometrical modeler is THRED, a free form surface modeler [6]. In this modeler the

surfaces are represented by polygonal meshes. The same techniques could be used to create and modify spline based surfaces, but polygonal meshes were chosen to simplify the implementation, and they can be displayed much faster in VR applications. The modeler is based on an hierarchical surface representation, similar to that used by Forsey in his Dragon editor [2]. In this representation, the user can select a part of the surface and produce a finer polygonal mesh to control the details of the surface's shape. Editing operations can be applied to the finer mesh to control the detailed shape, and to the coarser grid to control the overall shape of the object. A hierarchy of meshes can be constructed, with different parts of the surface having different levels of refinement, to give the user a range of controls over the object's shape.

The editing operations that can be applied to the surface are contained in a ring menu. These operations include dragging part of the surface and a number of different deformation techniques. One of the interesting aspects of its user interface is the user of two 3D trackers, one in each hand. This allows the user to select commands and provide context with one hand, and perform the actual editing operations with the other hand.

We are presently evaluating the interaction style used in this geometrical modeler to show the effectiveness of two handed interaction. Three versions of this geometrical modeler have been produced, each with a different user interface. One version uses the two handed style of interaction described above, and the second uses a single Bat and function keys to perform geometrical operations. The third version is a mouse based user interface to the same functional core used in the other two modelers. We hope that this will allow us to determine the effect of user interface style on designer productivity.

5. Conclusions

This paper presents a number of interaction techniques that can be used in 3D geometrical modelers. It also briefly describes three geometrical modelers that employ these interaction techniques. Our preliminary work indicates that providing 3D interaction and display techniques improves designer effectiveness.

References

1. J. Butterworth, A. Davidson, S. Hench and T. Alano, 3DM: A Three Dimensional Modeler Using a Head-Mounted Display, *Proc. 1992 Symposium on Interactive 3D Graphics*, 1992, 135-138.
2. D. Forsey and R. Bartels, Hierarchical B-Spline Refinement, *Proc. Siggraph'88*, 1988, 205-212.
3. S. Halliday and M. Green, A Geometric Modeling and Animation System for Virtual Reality, *Proc. VRST'94*, 1994, 71-84.
4. J. Liang and M. Green, JDCAD: A Highly Interactive 3D Modeling System, *Computer and Graphics 18*, 4 (1994), 499-506.
5. E. Sachs, A. Robers and D. Stoops, 3-Draw: A Tool for Designing 3D Shapes, *IEEE Computer Graphics and Applications 11*, November (1991), 18-24.
6. C. Shaw and M. Green, Two-Handed Polygonal Surface Design, *Proc. UIST'94*, 1994, 205-212.
7. C. Ware and D. Jessome, Using the Bat: A Six Dimensional Mouse for Object Placement, *Proc. Graphics Interface'88*, 1988, 119-124.