3DM: A Three Dimensional Modeler Using a Head-Mounted Display

Jeff Butterworth, Andrew Davidson,
Stephen Hench and T. Marc Olano

Department of Computer Science, Sitterson Hall
University of North Carolina
Chapel Hill, NC 27599-3175

1 Introduction

The use of interactive three dimensional (3D) environments has increased the demand for complex 3D models.[Sproull-90] The 3D environments that provide a sense of telepresence or "virtual reality" require a large number of models in order to give the user the illusion of being in a specific place. This demand for more models has highlighted the fact that most modeling systems are difficult to use for all but a small number of experts.[Sproull-90] Through identification of some of the fundamental obstacles to modeling we hope to make it accessible to more users.

The nature in which objects are typically selected and displayed is a major hindrance to 3D modeling.[Brooks-88] To specify the position of an object in 3D requires six parameters, the position (three) and the orientation (three). Most modeling systems (modelers) must settle for a 2D mouse augmented by a keyboard for this purpose. This mismatch results in difficult placement and picking of objects in modeling space. The display of models usually takes the form of a projection onto a 2D monitor. This has the effect of making spatial relationships unclear. Technological improvements to 3D model display and manipulation hardware can remove these barriers to model creation and understanding.

Current virtual reality technology provides one solution to intuitive modeling. A head-mounted display (HMD) system gives the ability to understand complex spatial relationships of models by placing the user in the model's world. Within this type of system, a hand-held pointing device supplies users with the ability to specify 3D relationships through direct 3D manipulation. An HMD system can be used to solve one of its own basic problems; the user can build the virtual world from *within* the virtual world. However, modeling does not instantly become simple when the user puts on an HMD.

The complexity of most 3D modelers' user interfaces presents another obstacle to efficient modeling. A source of this complexity is the large number of sophisticated commands and modes that can be used in current modeling systems. These features are often accessed by typing commands or by selecting options from large menus. Current virtual reality technology renders this type of control impossible. HMD based systems preclude use of a keyboard, and their low resolution makes it difficult to read menus of text.

Our source of inspiration for designing a user interface for an HMD-based modeler is the current software used for 2D modeling. At one time, creating 2D models required cumbersome CAD programs. This software took a long time to learn and often did not provide real-time interaction. 2D drawings are now easily manipulated by even the most casual users of personal computers. This revolution is in part the result of intuitive drawing programs like MacDraw. MacDraw represents an excellent example of how to make modeling more accessible to the common user. One of the keys to MacDraw is simplicity. Most work done with it requires no reading or use of the keyboard. MacDraw provides a palette of tools which is always available next to the model. To change modes, one simply uses the mouse to select from this palette. The process of 3D modeling can become more accessible if some of the lessons learned in this simplification of 2D modeling can be applied to 3D modeling systems.

This paper presents an HMD based system called 3dm which simplifies the task of 3D modeling by implementing the concepts introduced above. The design of this modeling system is explained below in order to understand the choices made in the implementation. The implementation of 3dm is described through a presentation of the most useful features. Finally, the results of actually using 3dm are presented with an emphasis on new techniques that can be applied within other virtual worlds.

A large body of work has been done on 3D modeling, but very little has been done with an HMD. Modeling using an HMD system has been explored by Clark.[Clark-76] Users of this system created parametric surfaces by manipulating control points on a wire-frame grid. This system highlighted the utility of using an HMD based system for improved understanding and interaction with models. 3dm relies on an HMD to help simplify modeling, but the intuitive user interface design is necessary to making 3dm easy to learn and use.

2 Design

3dm is an HMD based 3D modeling system employing proven user interface principles from 2D drawing systems. The fundamental goal of 3dm is to be intuitive and easy to learn for users with no prior modeling experience. To accomplish this task, 3dm has been designed with two principles in mind: bringing the user and the model together in one work space and giving the user easy access to modeling tools. The first principle is primarily a consequence of using an HMD, but the second requires more planning to implement.

Providing easy access to all of the features of 3dm is solved through the use of a *toolbox*. The toolbox is similar to the MacDraw tool palette, but in addition to supplying a visual method for changing modes, the 3dm toolbox must also be the method of access to all operations. The 3dm toolbox must always be visible and within reach and the function of its contents must be obvious. Details of the toolbox implementation are in the next section.

Although many modeling systems are considered interactive, 3dm must be especially responsive because the user is immersed in the modeling space. It is imperative that users not experience long pauses between screen updates while a slow operation is performed. Long or uneven pauses between screen updates have been shown to cause confusion and nausea for many people.[Deyo-89] 3dm meets the need for interactive reactions during all operations. For example, when the user scales a model, it grows and shrinks interactively rather than snapping to a new size after the operation.

3dm is designed to be easy to use, but we also want to support many familiar 3D modeling operations. It is hoped that traditional modeling operations will become easier to use, and thus more powerful, when implemented with the 3dm framework. Some of the power of pre-existing modelers comes from surface creation features like surfaces of extrusion and rotation. With this in mind, 3dm has been designed to support a variety of surface creation paradigms. Tools for instancing and editing hierarchical models have also shaped the design of 3dm because of their utility for animating and reusing models. Another class of useful tools are those found in simple drawing programs, like cutting and pasting in MacDraw.

The purpose of 3dm is for real users to create models, but it also offers an opportunity for learning about what makes a modeling system useful. Our design reflects this philosophy, making it possible to easily modify 3dm to fulfill users' requests. Ease of modification encompasses the ability to modify global behavior without large changes to the source code and the ability to add new tools and commands in a modular fashion.

3 Implementation

3dm was developed using a VPL eyephone as the display device and Polhemus trackers to track the head and hand. The 6D 2-button mouse was developed at UNC-CH. The images were rendered using the Pixel Planes 4 and Pixel Planes 5 high performance graphics engines developed at UNC-CH.[Fuchs-85][Fuchs-89] Currently all models created with 3dm are made up of hierarchical groups of triangles.

3.1 User Interface

In addition to the model, the virtual world of 3dm contains the components of the user interface. The user stands on a "magic carpet" which marks the boundaries of where the tracking system operates. Below the magic carpet lies a checkered ground plane on which the model is usually created. The cursor and the toolbox are the only other items that are always visible. The cursor follows the position of the hand-held mouse.

The toolbox initially appears suspended in space near the user's waist. The user can grab the toolbox and move it to a more convenient location. The toolbox remains attached to the user as he or she moves around the modeling space. The toolbox can be disconnected from the user and left anywhere above the magic carpet and can be reconnected at any time. The toolbox is organized into cells containing 3D icons. Each icon represents either a tool, command, or toggle. These icons perform actions when they are selected with the cursor. Tools change the current mode of operation which is reflected in the shape of the cursor. For instance, when the user reaches into the toolbox and selects the flying tool, the cursor takes the form of an airplane. Selecting a command performs a single task without changing the current mode of operation. Toggles do not change the current mode of operation or immediately perform any task. Instead, toggles change some global aspect of 3dm. An example is the snap-to grid toggle, which restricts cursor movement to a 3D grid when it is on.

Exploring the model provides understanding of its 3D shape, so 3dm supports multiple methods of navigating in the modeling space. The HMD system used for 3dm allows the user to walk through the model space a few paces in any direction. 3dm supports "flying," a commonly used method of traveling through virtual worlds, which consists of translating the user through model space in the direction that the cursor is pointing. Flying moves the magic carpet, which carries the user and the toolbox along. A method of navigation that is the complement of flying is grabbing the world. Grabbing the world allows the user to attach the modeling space to the cursor and then drag and rotate it. Grabbing can be used to bring a feature of the world to the user rather than forcing the user to walk or fly to the feature. Models often require manipulation at vastly different scales. To facilitate this type of work, the user can be scaled using a process called *growing* and *shrinking*. This scaling does not affect the model: it changes the user's relative size with respect to the model. The user could shrink down to bird size in order to add eyelashes to a model of an elephant and then grow to the size of a house to alter the same model's legs.

The user receives continuous feedback in a variety of ways. The HMD system provides all visual input to the user, so the display must be updated between 15 and 30 times per second. Even during file loading and other slow operations, the screen is updated and the head is tracked. Rubber banding is implemented in many situations: when defining a new triangle, scaling or moving an object, and extruding. Predictive highlighting shows the user what *would* be selected if a mouse button was pressed. This highlighting is used in the toolbox, but even more importantly it is used when marking vertices.

3.2 Tools and Commands

Although many tools are available in the 3dm toolbox, it is more useful to understand the general classes of tools supplied to the user than to enumerate all of the specific tools. Most of these tools were chosen because of their proven utility in pre-existing modelers.

Surface creation is the central purpose of most 3D modeling, so 3dm provides more than one method for creating surfaces. A triangle creation tool exists for generating both single triangles and triangle strips. Pre-existing vertices may be used during triangle creation to allow triangles to share corners or entire edges, making seamless connections easy.

The extrusion tool supplies a more powerful but more specialized method of triangle creation. This tool allows the user to either draw a poly-line or select one from edges already in the model and stretch it out into an extruded surface. The extrusion is performed by dragging the leading edge of the surface with the mouse. Because the mouse can be twisted and moved arbitrarily during the extrusion, it becomes easy to create complex surfaces with this tool. In addition, the leading edge of this new surface can be scaled and then extruded again as many times as necessary. This form of extrusion can create such objects as walls, legs, tree trunks, or leaves in a matter of seconds.

The last set of surface creation tools facilitates creation of standard surface shapes. Currently box, sphere, and cylinder tools exist. They each allow the user to stretch out an arbitrarily proportioned wireframe representation of a standard shape. When the wireframe representation has the desired proportions it turns into a triangulated surface.

Since surfaces are rarely in exactly the desired shape upon creation, it is important that surface editing be an easy operation. The tool that marks and moves objects is the most commonly used tool. Not only can entire objects be grabbed and moved with the mouse, but selected groups of vertices can be moved in order to distort part of an object. Scaling can also be performed on either entire objects or groups of vertices. During both movement and scaling the user sees the model changing in real time. This interaction decreases the number of edits needed to make a desired change.

Familiar editing operations from drawing programs are a group of 3dm commands that facilitate rapid experimental changes. An arbitrary number of triangles or entire objects can be cut, copied, pasted and deleted. These commands provide easy reuse of existing objects. An undo/redo stack is provided for reversing any number of operations from any tool or command.

The hierarchical features of 3dm provide methods for organizing complex models. Grouping, like in MacDraw, can be used to associate triangles and possibly other groups in order to more easily manipulate them as a whole. The ability to instance these groups and impose a hierarchy on them can be used to organize the model further.

3.3 Experimental Framework

One of the basic design decisions for 3dm is that it should be easily changed to experiment with altered global behavior and tools. Most of this goal is accomplished through the use of traditional modular program design. However, some aspects of it are worthy of elaboration.

Similar to widget based toolkits in the X Window system, 3dm has a central event dispatcher. Basic features like cursor tracking, screen updating, and input handling are controlled by this dispatcher. The event dispatcher maintains a table that contains function pointers for each tool. These function pointers are callbacks for the various input events. Thus, the source code for a single tool can be isolated in a file containing its reactions to approximately 10 standard event types.

Database management functions encapsulate low level database modification. Any modifications or queries of the geometry database must go through the database management functions. Abstraction of database manipulation means that the internal database implementation can be changed without altering existing tools.

4 Results

Actual modeling sessions have shown that 3dm is efficient for rapidly prototyping models. Organic shapes, like rocks and trees, have proven to be particularly good subjects for 3dm. These shapes are easily created in 3dm because it provides a good sense for spatial relationships. Users of 3dm have commented that they feel a sense of control, because they can reach out and grab any part of the model with ease. The ability to make these quick modifications encourages the user to experiment with a shapes until they are satisfactory. However, 3dm has shown weakness in the area of constraints and models that traditional CAD and drawing programs create well. 3dm does not yet have enough methods for constraining users' movements.

The extrusion tool is an example of a traditional modeling tool that has become even more powerful because of its use in an HMD framework. In most modeling systems extrusion is usually performed by moving one or two spatial parameters at a time. 3dm users often alter many parameters at once during an extrusion by twisting *and* translating the new surface. Extrusion in 3dm often consists of many short extrusions. In between these short operations the leading edge of the extruded surface is often scaled and twisted. The result is that complex surfaces can be rapidly created with this tool that remains easy to use.

As an experimental framework, 3dm has proved quite useful. New tools have been added and global behavior has been modified to satisfy user requests. Early on, users complained about the screen freezing during file loading. As a result, we implemented a "busy" clock that replaces the cursor during slow operations. Users also commented that it was difficult to perform a "click" (rapid button press and release) because 3dm often interpreted it as a "drag" (button press, movement, then release). This problem was easily solved by altering the way in which the event handler differentiated between clicks and drags.

The ability to add new global behavior has begun to solve the lack of constraints in 3dm. Some initial solutions have been to add toggles in the toolbox for a snap-to grid and a snap-to plane. The snap-to grid constrains the position of the cursor to the nodes of a regular 3D grid. The resolution of the snap-to grid is dynamically modified to be appropriate to the user's current "grown" or "shrunk" size. The snap-to plane gives the ability to constrain cursor movement to 2 dimensions.

Tools have been added to 3dm easily because of the isolation of each tool's code from the rest of the system. The standard surface creation tools are examples of tools that were not part of the original 3dm implementation. They were added after users requested them and their implementation required only one new file of source code and a new entry in a global callback function table.

5 Conclusion

3dm has drawn techniques of model manipulation from both CAD and drawing programs. These familiar techniques have become even more powerful because of their increased ease of use within the 3dm framework. Through the extensible nature of 3dm, we plan to add more modeling constraints in order to make more regular models, such as buildings and machines.

3dm has been shown to solve the problem of making 3D modeling accessible to unsophisticated users. It takes advantage of users' natural forms of interaction with objects to give them better understanding of the shapes of their models. Modeling with 3dm encourages users to experiment with model shape because it is easy for users to make rapid changes.

References

[Brooks 88] Frederick P. Brooks, Jr. Grasping Reality Through Illusion: Interactive Graphics Serving Science. *Keynote address at the Fifth Conference on Computers and Human Interaction.*, Published in CHI '88 Proceedings, May 1988, 1-11.

[Clark 76] James H. Clark. Designing Surfaces in 3-D. *Communications of the ACM*, 19(8):454-460, August 1976.

[Deyo 89] Roderic Deyo, David Ingebretsen. Notes on Real-Time Vehicle Simulation, Chapter 4. In *Course Notes: Implementing and Interacting with Real-Time Microworlds.* ACM SIGGRAPH, 1989.

[Fuchs 85] Henry Fuchs, Jack Goldfeather, Jeff P. Hultquist, Susan Spach, John D. Austin, Frederick P. Brooks, Jr., John G. Eyles, John Poulton. Fast Spheres, Shadows, Textures, Transparencies, and Image Enhancements in Pixel-Planes. *Computer Graphics*, 19(3):111-120, February 1985. SIGGRAPH '85 Conference Proceedings.

[Fuchs 89] Henry Fuchs, John Poulton, John Eyles, Trey Greer, Jack Goldfeather, David Ellsworth, Steve Molnar, Greg Turk, Brice Tebbs, Laura Israel. Pixel-Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories. *Computer Graphics*, 23(3): 79-88, July 1989. SIGGRAPH '89 Conference Proceedings.

[Sproull 90] Robert F. Sproull. Parts of the Frontier are Hard to Move. *Computer Graphics*, 22(2):9, March 1990. Symposium on Interactive 3D Graphics.