Non-WIMP Research at University of Alberta

Chris Shaw
615 General Services Building
Department of Computing Science
University of Alberta
Edmonton, Alberta
CANADA
T6G 2H1

Phone (403) 492-7434
Fax   (403) 492-1071

1. Introduction

At the Computer Graphics lab at the University of Alberta, we have been experimenting with "HIGGS" user interfaces, for Headmounts, Isotraks, Gloves, Gestures, and Stereo. That is, we have built a number of simple demonstration programs that illustrate the use of the DataGlove, the Isotrak, and the Eye-Phone head-mounted display.

The demonstration programs are as follows:

1) A simple mesh viewer uses an Isotrak to manipulate the position and orientation of polygonal meshes.

2) A behavioural simulation of a fish tank simulates fish motion. The fish swim together, and they avoid the DataGlove, with which the user tries to grab a fish.

3) A simple architectural simulation uses the EyePhone to view a static model of some rooms.

4) A fluid dynamics simulation uses the EyePhone to view the surface of the fluid, and the DataGlove to input force boundary conditions on the surface.
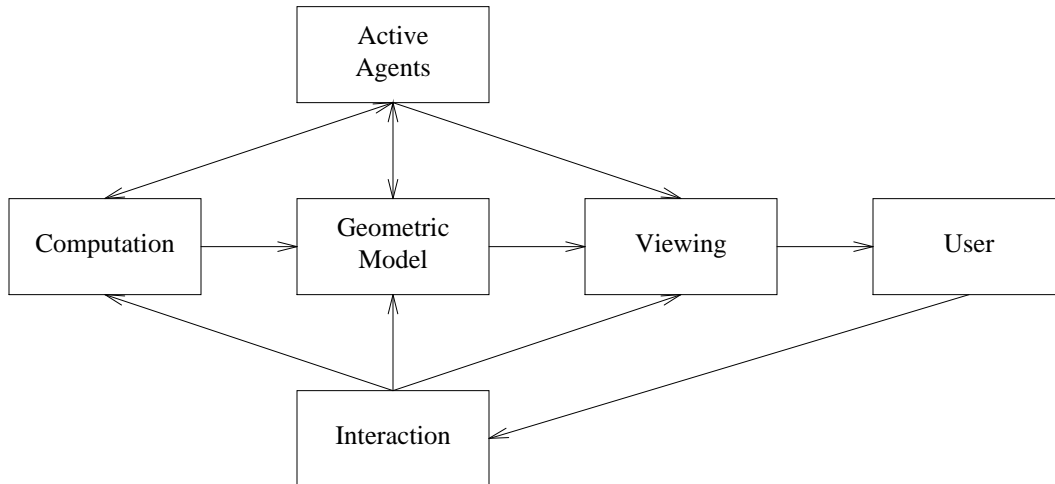

Figure 1: Model of 3D Interaction

These demonstrations have pointed out the need for software tools of various kinds, since there are problems that must be solved that are common to each demonstration program. Figure 1 shows a model of 3D interaction proposed in [Green90] which illustrates the common structure of the above demonstrations. This model views the user as manipulating a software system via an interaction, and viewing its results. The Interaction component manages the Computation, Geometric Model and Viewing components

according to the user's wishes. The Active Agents also manipulate and observe the Computation, Geometric Model and Viewing, although it is anticipated that the agent and the user are not working at cross-purposes. The agent could be another user or group of users, in which case this model is a structure for computer-sponsored co-operative work.

The Computation element could be a scientific simulation, the controller of a remote manipulator, the behavioural animation of some fish, as in demonstration 2, or the Computation component may not exist. The Geometric Model component is always present. In CAD applications, the Geometric Model is the object of interest, and is being edited, while in a scientific simulation, the geometry is simply a means of viewing the behaviour, as in demonstration 4. The Viewing component encompasses not only the appearance of the chief object or objects under consideration, but also the administrative things such as the 3D cursor, menu appearance, and so on.

We have been working on software structures to support this model of interaction. In particular, the middle-level software tools mentioned in section 5 implement the Modeling and Viewing aspects, while the the high-level tools provide the Interaction and Active Agent tasks.

This structure helps us focus on the sorts of tools required to provide effective 3D user interfaces. There are four levels of tools. The tools at the bottom level are responsible for driving the various 3D I/O devices. Next, at the low level are tools that deal with workspace installation and device calibration. At the middle level are tools for the design, modeling and viewing of objects, and at the high level are tools for specifying behaviour of the user interface. The following four sections will discuss the tools at each level. Section 6 will propose a metaphor for the design of 3D user interfaces.

2. Bottom-Level Tools

The bottom level tools are programs which drive the 3D input devices. There are two main device types: the Polhemus Isotrak, and the VPL DataGlove. The VPL EyePhone contains an Isotrak for tracking head position, and is driven by a pair of NTSC sources.

Each driver is based upon the client-server model. The server program runs continuously on a network-connected host, and accepts client requests from user programs. When a user program starts up, it connects to the servers for the devices it wants, and then polls the servers for device data. The server not only performs direct device control, but can also provide higher-level functions such as data filtering.

There is a third software package that automatically sets up the co-ordination of stereo viewing. The user provides a display routine, and the viewing package takes care of display co-ordination between the two graphics workstations that drive the head-mounted display.

These tools work well as a first cut, but the interaction is less than satisfactory due to device lag, and due to the fact that the host operating system does not treat these devices as first-class citizens.

3. Low-Level Software

The Isotrak, DataGlove, and EyePhone each must be calibrated in some way. In the case of the Data-Glove, this means that as a user wears the DataGlove anew, the DataGlove must be calibrated such that the rendering of the user's hand on the screen closely matches the hand's actual shape. For the EyePhone, the user's inter-ocular distance must be measured so as to give accurate stereo images. Both of these calibrations can be saved, and used again when the same user returns, with a small loss in accuracy. We are experimenting with tuning techniques to make saved DataGlove calibrations more accurate.

The Isotrak implements its own co-ordinate system, whose origin depends on the position of the source module. We have developed a low-level tool that maps from Isotrak device co-ordinates to real-world room co-ordinates to application co-ordinates. This triple mapping resides in a configuration file that is read upon application startup.

4. Middle-Level Tools

At the middle level, we are developing tools and techniques for modeling and viewing. This corresponds to the Geometric Model and Viewing components in Figure 1.

The first aspect of modeling is that in some application domains such as scientific visualization, a

means must be found to convert the scientific data into some displayable format. While it may be tempting to have the scientific program output polygons, this limits the type of visualization possible, and no doubt increases the sheer quantity of data sent from Computation component to the Viewing component. More-over, this type of modeling limits the available interaction, since the geometry that the user sees is also all that the interaction can manipulate. We are working on a modeling tool that allows the user to create parameterized primitive objects as simple as a polygonal mesh, but which will also allow the user to extend the primitives to include other information such as colour, mass, moments of inertia, density, sonic feed-back, highlight colour and so on. The primitive objects can then be combined in a hierarchical manner to create the high-level objects of interest.

The advantage of extensible modeling is that the user interface should be able to manipulate not only the geometry, but also the extended data. In addition, simulation programs such as scientific code can also manipulate the same data in this common format.

Viewing is also an important aspect. Interaction with a 3D model requires at least a 10 Hz update rate, and the modeling environment should enable this, since a parameterized geometric model can also supply viewing information, such as a BSP tree. We are working on methods to extend the BSP tree algo-rithm to display dynamic models.

5. High-Level Tools

Currently, the high-level tools are at a conceptual stage, so this section will outline the purpose of these tools. The most important tool is the Volume Manager, which is the 3D equivalent of the 2D window manager. This tools manages the allocation of the real 3D space that the user resides in, and assigns a con-text for interpreting device input based upon where the devices are. So, for example, if the user is about to use the mouse, the Volume Manager can decide to ignore all DataGlove activity when the DataGlove is in the volume enclosing the mouse and mouse pad. Specialized contexts of this nature can solve the problem of accidental gesture recognition. A special context can also provide the equivalent to the McIntosh menu bar, such that when the DataGlove is in this special volume, menu items appear.

The Volume Manager should also handle picking, dragging, and similar operations upon the objects within the Geometric Model. Typically there is a problem with adequate feedback of cursor position and orientation (in the absence of force feedback), and the Volume Manager tool should handle this.

A third aspect is navigation through simulated 3D spaces that are much larger than the space that the user occupies. For example, in an architectural viewing application, the user wants to be able to move from room to room in without exceeding the physical range of the EyePhone's harness. Interaction techniques or other devices must be developed to allow navigation. We have built a small belt-mounted knob-and-button box called "the navigation box" for this purpose, and we are working on interaction techniques that allow the user to control a virtual vehicle that navigates about large virtual spaces.

The other high-level tool is the active agent manager. A user may program an active agent to observe a particular subset of the Geometric Model during a scientific computation, for example. The agent manager must be able to cleanly interface to the user's view when it spots something interesting. More-over, if the agent is a person, the other person may want to temporarily control the user's view, so the agent manager must be able to allow this in an ergonomic manner.

6. A Metaphor of 3D Interaction

Historical accounts of human memory skills usually refer to the medieval Christian monks, who were able to memorize large chunks of text through the use of spatial visual cues. Saint Thomas Aquinas is said to have imagined his memory as a mind-castle, in which each room contained some of his knowledge. To remember the knowledge, Aquinas pictured himself in the room, which gave him the spatial visual cue needed to remember the memorized text. Of course, the monks needed this memory skill because they had to make their books by hand.

However, this "Buildings" metaphor may be useful for building 3D user interfaces, since putting up walls is a means of allocating and claiming space. Since the fundamental resource in 2D user interfaces is screen space, and since 2D UIMS's manage and allocate that space in some consistent metaphoric manner, it seems clear that the fundamental resource for 3D user interfaces is volume. If a window contains a 2D

context, then maybe a room ought to contain a 3D context, much in the same way that real rooms contain real contexts, such as "kitchen", "dining room", "main office" and so on. Thus, and a 3D user interface, one may have a whole building devoted to one's scientific visualization activity, while across the virtual street in the party building there's a game of 3D "rogue" in progress.

Fortunately, these metaphorical buildings are not constrained by physical laws, so they can be of any size and shape, and rooms may themselves contain whole campuses devoted to some very specialized context. Of course, rearranging the furniture is simple, as is rearranging the structure of a room, its hallway, its floor, and its building.

The computational advantages are that the graphical workstations rendering the pictures need only render the contents of the room, and need only store the room's contents in memory.

7. Conclusion

We have presented a summary of past, present, and future work on 3D user interfaces at the University of Alberta. This includes a model of 3D interaction for a wide variety of applications, and a metaphor for the design of future coherent 3D user interfaces.

Green90.M. Green and C. Shaw, The DataPaper: Living in the Virtual World, *Graphics Interface 1990*, Halifax, Nova Scotia, May 1990, 123-130.