CHAPTER I

INTRODUCTION

1.1 Motivation

Immersive virtual environments (VEs) made their debut in the late 1960s when Ivan Sutherland created the first system involving a tracked head-mounted display (HMD) and real-time three-dimensional computer graphics (Sutherland, 1968). The system was crude, and the amount of computing and rendering power was minuscule, compared to today's technology, but all of the basic components that make up the virtual reality (VR) systems of the 1990s were present in Sutherland's prototype.

Since that time, there have been over thirty years of continuous research in the area of virtual environments. New hardware technology is continuously in development that allows us to render more complex 3D scenes at interactive frame rates. Graphics displays have seen tremendous improvement: we are able to display millions of different colors simultaneously on a very large screen at a refresh rate so fast that the human eye cannot perceive the flicker (Foley et al, 1990). There are many different tracking technologies available which provide 3D position and orientation data for multiple receivers simultaneously (Meyer and Applewhite, 1992). Technologies are being developed which provide input to other human sensory modalities besides vision. Haptic devices allow a VE user to seemingly "touch" virtual objects (Gomez, Burdea, and Langrana, 1995). Spatial sound creates the illusion of audio sources coming from certain locations in the 3D space (Durlach, 1991). There is even research into the use of olfactory input in virtual environments (Dinh et al, 1999).

VE research has not focused entirely on hardware; software advances have also been made. Algorithms have been implemented and refined in the areas of model simplification, level of detail culling, geometry database management, texture mapping, lighting and shading, hidden surface elimination, and so on. All of these algorithms allow us to present a more complex and more realistic environment, while still maintaining real-time frame rates. Also, large software systems have been created expressly for the purpose of aiding the development of virtual environment applications (e.g. Kessler et al, 1998). These VE support systems can handle rendering, model maintenance, lighting, interfaces with trackers and other input devices, etc. This allows the developer to focus on the components which distinguish his VE application from others: the environment itself and the behavior of the application (e.g. response to button presses, animation, and interaction with virtual objects).

What does the virtual environment community (primarily university researchers, small commercial ventures, and hobbyists) have to show for these thirty years of advancement in hardware and software specifically targeted at immersive VEs? Certainly, the degree of realism and complexity has increased, and making the virtual world more believable in this way may lead to a higher sense of immersion, or presence, for the user. But what applications have emerged into more common use outside of the laboratory? Surprisingly, our experience in the field indicates that there are very few VE applications in common use. To understand why, we should examine those applications that have become useful, and determine their common characteristics that allowed their success. Three such applications are architectural walkthrough, psychotherapy, and VE gaming (we discuss flight simulation and training, two other applications used for real work, below).

Architectural walkthrough (Brooks, 1992) was perhaps the application which brought VE technology into the public eye more than any other. The basic idea is simple: the user can be immersed within a 3D model of an architectural space, and view it and move about it from a first-person perspective, as she would in an actual building. In this way, architects can verify the appropriateness and visual impact of their designs, engineers can study physical aspects of the space, and prospective clients can assess the current status of the project and suggest changes before a structure is even built. Why are VEs needed for this task, rather than simply viewing 3D models on a computer screen? One possible reason is that the user is immersed within the model, and can use her proprioceptive and kinesthetic senses to evaluate the space in a natural manner. Furthermore, this application requires only one additional component over those first proposed by Sutherland: some method of moving the user's viewpoint about the space.

Applications in the field of psychotherapy (Hodges et al, 1995, North, North, and Coble, 1996) have emerged rapidly since the early 1990s. One of the most well-known areas, which is beginning to see practical usage, is the treatment of various phobias. A common method of therapy for phobias is called graded exposure. The patient is placed in a situation in which the fear is triggered, but only slightly. He remains there with the therapist until he has mastered his fear in that situation, at which point a slightly more intense situation is presented. In this way, the patient gradually becomes able to deal with his fear. For example, to treat acrophobia, the fear of heights, the patient might be taken to a second floor balcony, then a fifth floor balcony, then the roof of a ten story building. This treatment has been shown to be effective, but also time-consuming, potentially embarrassing for the patient, and sometimes costly. The only requirement for exposure therapy is that the patient feel present in a situation which triggers his fear, which makes this application a natural one to try in a VE. The treatment can now take place in the therapist's office, without the time, embarrassment, or cost associated with traditional exposure therapy. Unlike architectural walkthrough, VE exposure therapy does not even require a means for the user to move about. It is usually sufficient for the user to be able to sense the environment and to look around (using head tracking), so that the fear stimulus can be perceived.

VE entertainment and game applications have also become popular in recent years. This has most often taken the form of location-based entertainment (LBE) through companies such as Virtuality™, which involves a complete VE system installed in some permanent location, with users paying for each game. In any case, most of the games available for such systems can be characterized as first-person "shoot-em-up" games, in which the user moves through the virtual environment shooting his enemies. In many ways, the requirements of these games are similar to those for architectural walkthrough: real-time 3D graphics, head tracking, and some technique for moving through the environment. The only additional requirement is some sort of weapon that can be aimed and fired at the enemies in the game.

What do these applications have in common? It seems that they all benefit from the enhanced sense of presence that an immersive virtual environment provides. "Being there" is what makes these systems more compelling or useful than the same 3D graphics rendered on a screen, with no head tracking. However, we also claim that each of these applications requires very little in terms of user interactivity. In applications such as exposure therapy, the user is mostly passive, simply looking around the space using standard head tracking. In the walkthrough and entertainment applications, the user may be more active (moving through the space, shooting, etc.), but the actions are very simple and repetitive. We would call this a high frequency but a low complexity of interaction.

There are, however, a small number of applications being used for real work which have more complex characteristics of interaction. These include flight and vehicle simulation, which has been in use for many years, and training applications such as those used by NASA for simulation of astronaut “space walks.” Although these applications are more complex, the interaction is designed in a manner very specific to the system, and not in a way that could be extended to other types of applications. As Fred Brooks pointed out in his 1999 keynote address to the IEEE Virtual Reality conference, this is most often done by replicating the devices that the user would interact with in the real-world situation (e.g. the throttle and flight stick, or the spacesuit controls) and using those to drive the simulation. Because of this specificity to the application domain, we claim that there is little that we can learn in general about VE interaction from such systems.

On the other hand, many more application areas have been proposed and researched for immersive VEs. The architectural community wants to take the walkthrough to the next step and be able to not only view, but also design artifacts in a VE (Bowman, 1996, Mine, 1997). Prototype scientific visualization applications have been developed (Bryson and Levit, 1992, Taylor et al, 1993), in which scientists can interactively view complex simulations and structures, and also change the parameters of the simulation, move and regroup elements, and so on. Educational applications have been proposed (Dede, Salzman, and Loftin, 1996) that allow students to learn about certain concepts by engaging themselves in a virtual laboratory, and viewing the effects of changes first hand. The list goes on.

However, we have not seen these applications in common use. It is our opinion that this is not because they are inappropriate for immersive VEs, but because their requirements for interaction are much more complex than the applications discussed previously. These systems require not only head tracking and a method of movement, but also the ability to select objects, to pick up, position, orient, and place objects, to change the system mode, to control the speed of a simulation, etc. One could argue that these applications are not in the mainstream due to the limitations of technology (input devices, trackers, displays, etc.), but researchers have been attacking the technology problem for thirty years. Our claim, on the other hand, is that because little research has been devoted to the *user interface* and *interaction techniques* for immersive VEs, the resulting prototype applications are not as usable as they need to be, and therefore do not see real-world usage. We must ask the question, “Given the current state of VE technology, is it possible for a virtual environment system to simultaneously be immersive, have complex interaction, and exhibit high levels of usability?”

Why is it difficult to develop appropriate user interfaces and interaction techniques for immersive virtual environments? Shouldn’t interaction in VEs be completely natural, replicating the real world? Some have argued that this should be the case (Nielsen, 1993). Considering the applications we wish to develop for VEs, however, such natural interaction would be woefully inadequate. Instead, we want to extend the user’s physical, perceptual, and cognitive capabilities so that real work can be performed in a VE that could not be done easily in another setting. Therefore, we need new techniques for interaction.

Why is the current state of the art not good enough? Interaction research (human factors, human-computer interaction, user interfaces, etc.) is almost as old as the first computers. Many usable applications have been developed for the traditional desktop metaphor which have extremely complex interaction requirements. However, interaction in immersive VEs faces many difficulties that make it not only harder to develop, but also fundamentally different, than traditional user interfaces.

Desktop interfaces are inherently more constrained than immersive interfaces. Most desktop applications use only two dimensions, which map directly to the 2D control of a mouse. The mouse rests on a surface, so it does not have to be held continuously by the user, and this allows the user to position it very accurately. Text entry is simple and standardized with a keyboard. On the other hand, input devices for immersive VEs are generally three-dimensional, and must be held in place continuously, resulting in lower accuracy. Tracking devices also have inaccuracies, as well as latency which causes the displayed image to lag behind the actual tracker positions. Text entry is generally extremely difficult or impossible, because the user cannot use a standard keyboard while wearing an HMD and/or holding other input devices. Besides these problems, most common HMDs have lower resolution than monitors, so that screen space is even more valuable.

All of these difficulties combine to make usable immersive interfaces much more problematic to design than their desktop counterparts. This is not to say that all previous user interface research is invalid for immersive VEs. On the contrary, certain high-level guidelines and concepts (e.g. Norman, 1990) apply perhaps even more to VEs than traditional systems, because the user interface must be even more transparent and intuitive in order to overcome the other limitations. However, because of the fundamental differences between traditional and immersive interfaces, new research is required that focuses solely on interaction in immersive VEs. Indeed, in his 1999 IEEE Virtual Reality keynote address, Dr. Fred Brooks stated that finding the best ways to interact with virtual environments was one of the five most important open questions in the field.

In this work, therefore, we are taking initial steps in a research program to develop an understanding of interaction techniques and user interfaces for immersive virtual environments. The goal will be both a qualitative understanding, as in user interface guidelines, as well as a quantitative model of performance and usability. Our contribution will be to evaluate and analyze the most common interactive tasks required by VE applications, as well as to categorize and evaluate various interaction techniques designed for these tasks. We will show the effectiveness of our evaluation by applying the results to an application designed for real-world usage.

1.2 Definitions

Before beginning our discussion of interaction techniques for virtual environments, it is important that we define each of the major terms that relate to this work, so that the boundaries and components of the problem are well understood. Some of these terms have disputed definitions, and we do not claim to offer the final word on these terms. We simply intend to provide definitions that allow the reader to understand the use of these terms in this thesis. The terms that we define here are relevant to virtual environments, user interfaces and interaction, and important technologies used in VEs.

- **Virtual Environment (VE)**: A three-dimensional model of a space displayed to a human user from an egocentric point of view using real-time 3D computer graphics. A single object model, viewed from the outside in, is not a VE by our definition. Motion and point of view orientation are generally controlled by the user, not the system. Thus, a first-person computer animation also does not qualify as a VE. VEs often include other sensory information, such as auditory or haptic cues.

- **Virtual Reality (VR)**: The experience of being within a VE. We prefer not to use this term, as it is associated with unrealistic hype and expectations portrayed in popular media.
- **Real-Time**: Displayed at a frame rate that ensures that images move smoothly as the view direction changes. The minimum frame rate that is considered to be real-time might be as low as 10 Hz, or as high as 30 Hz.
- **Immersion**: The feeling of "being there" that is experienced in some VEs. A VE user is immersed when he feels that the virtual world surrounds him and has to some degree replaced the physical world as the frame of reference. Immersion may take place in other media, such as films or even books.
- **Presence**: A synonym for immersion.
- **Immersive**: Surrounding the user in space. A VE is described as immersive when the computer-generated environment appears to enclose the user, and when the parts of the physical world that are not integral system components are blocked from view. In a head-mounted display (HMD), the graphics always appear on screens coupled to the user's head, but this produces the illusion that the VE surrounds the user completely. In a flight simulator, the graphics appear out the window, and are updated as the plane "turns" so that the VE seems to surround the user. The physical cockpit of the simulator is not blocked from view, but it is part of the simulation. For HMD or stereoscopic spatially immersive display (SID) systems, head tracking is required to make the system immersive.
- **Human-Computer Interaction (HCI)**: The exchange of information between human beings and computers during a task sequence for the purpose of controlling the computer (from the point of view of the human) or informing the user (from the point of view of the computer). This interaction usually has the goal of increasing human productivity, satisfaction, or ability (Hix & Hartson, 1993).
- **User Interface (UI)**: The hardware and software that mediate the interaction between humans and computers. The UI includes input and output devices, such as mice, keyboards, monitors, and speakers, as well as software entities such as menus, windows, toolbars, etc (Hix & Hartson, 1993).
- **Interaction Technique (IT)**: A method by which the user performs a task on a computer via the user interface. An IT may be as simple as clicking the mouse button, or as complex as a series of gestures. There may be many possible ITs for any given interaction task. The IT may be influenced by the input device used, but is separate from it. The same input device may be used for many ITs for the same task; conversely, it may be possible to implement a given IT using several different input devices.
- **Head-Mounted Display (HMD)**: A computer graphics display that is worn on the head of the user, so that the displayed graphics are continuously in front of the eyes of the user. HMDs may use Cathode Ray Tube (CRT) or Liquid Crystal Display (LCD) technology, and usually incorporate optical lenses to widen the

displayed image and move it farther from the user's eyes. Many HMDs include headphones for audio, and most are used in conjunction with trackers.

- **Spatially Immersive Display (SID)**: A computer graphics display which surrounds the user on more than one side. SIDs are usually implemented with rear-projection screens. Common SID types include the CAVE™ (Cruz-Neira, Sandin, and DeFanti, 1993) and dome displays. SIDs do not require the user to wear any headgear, except for stereo viewing glasses if stereoscopic graphics are used.
- **Tracker**: A device that measures 3D position, and sometimes orientation, relative to some known source. Common tracker types are electromagnetic, optical, ultrasonic, gyroscopic, and mechanical linkage (Meyer & Applewhite, 1992).

1.3 Problem Statement

How can we begin to analyze interaction techniques for immersive virtual environments? There are a multitude of tasks which one might conceivably want to perform within a VE, and most of them are application-specific. However, we can reduce the space of the problem by recognizing that there are a few basic interaction "building blocks" that most complex VE interactions are composed of. Such an approach is similar to that proposed by Foley for interaction in a 2D graphical user interface (Foley, 1979).

If, then, we can identify these universal tasks, understand them, and evaluate techniques for them, we will have come a long way towards understanding the usability and interaction requirements for immersive VE applications. From our experience with VE applications and discussion with other researchers, we have identified four task categories: *travel*, *selection*, *manipulation*, and *system control*.

Travel, or viewpoint motion control, refers to a task in which the user interactively positions and orients her viewpoint within the environment. Since head tracking generally takes care of viewpoint orientation, we are mainly concerned with viewpoint translation: moving from place to place in the virtual world. Selection is a task that involves the picking of one or more virtual objects for some purpose. Manipulation refers to the modification of the attributes of virtual objects, such as position, orientation, scale, shape, color, or texture. Selection and manipulation tasks are often paired together, although selection may be used for other purposes (e.g. denoting a virtual object whose color is to be changed). Finally, system control encompasses other commands that the user gives to accomplish work within the application (e.g. delete the selected object, save the current location, load a new model). We will not consider system control separately in this work.

For each of these universal interaction *tasks*, there are many proposed interaction *techniques*. For example, one could accomplish a selection technique in a very indirect way, by choosing an entry from a list of selectable objects. Alternately, one could use a direct technique, where the user moves his (tracked) virtual hand so that it touches the virtual object to be selected. Each of these interaction techniques has advantages and disadvantages, and the choice of a certain technique may depend on many parameters.

In general, we feel that interaction techniques for immersive VEs have been designed and developed in an ad hoc fashion, often because a new application had unusual requirements or constraints that forced the development of a new technique. With few exceptions, ITs were not designed with regard to any explicit framework, or evaluated quantitatively against other techniques. Currently, then, we have a large collection of ITs for VEs, but little in-depth understanding of their characteristics or analysis of their relative performance.

The goals of this research, then, are four-fold:

1. To develop formal characterizations of the universal interaction tasks and formal categorizations or taxonomies of interaction techniques for those tasks,
2. to use these characterizations to design new techniques for each of the universal tasks,
3. to develop and utilize quantitative experimental analyses for the purpose of comparing the performance of interaction techniques for the universal tasks, and
4. to show the validity of the formal frameworks and evaluations by applying experimental results to a real-world VE application which involves all of the universal interaction tasks.

1.4 Scope of the Research

A complete and thorough understanding of VE interaction and user interfaces is not a realizable goal at this point in the maturity of the research area. Therefore, in this work we will focus on specific pieces of the overall problem with high levels of importance and benefit to the VE community.

First, this thesis focuses on low-level interaction techniques – small methods that are used to carry out a single user task. We feel that VE interaction must be understood at this level before we can begin to discuss complete VE user interface metaphors. This is similar to the situation in 2D user interfaces when graphical UIs first became popular. The first step was to develop ITs that performed well and were easily understandable, such as push buttons, pull-down menus, windows, and sliders. Only when this was complete could these elements be combined to form a usable interface. This does not mean that we are neglecting the context in which interaction is performed; on the contrary this context is explicitly included in our design and evaluation framework. We simply desire to understand the components of a usable VE interface before proposing complete interfaces.

Second, this thesis assumes that the goal of interaction is a high level of performance. This may seem overly restrictive, but we take a broad definition of performance which includes not only time for task completion and accuracy, but also more qualitative measures such as ease of use, user comfort, and even the level of presence. Using this definition, almost any application can specify its interaction requirements in terms of performance metrics. However, there are cases in which the goal of a VE application is only loosely based on these performance metrics, such as a VE which simply attempts to replicate interaction in the real world (a naturalistic metaphor). Techniques such as these will not be considered in our design and evaluation.

Third, we choose to consider ITs for a small number of very common and important VE user tasks. Certainly, many interactive VEs contain tasks other than travel, selection, and manipulation, but these three seem to be the most universal and important to understand initially. Furthermore, many more complex interaction tasks are actually composed, at least in part, of these three tasks. Thus, we aim to identify techniques which produce high levels of performance on these generic tasks, so that these techniques can then be applied to the more specific tasks in an application. We do not claim that a general technique will always have better performance than one designed specifically for the task at hand (in fact, this may rarely be the case), but it is impractical to design a new interaction technique for each task in each application. At some level, interaction needs to be more general or even standardized.

Fourth, we are restricting our study to those techniques which are useful in immersive VEs. This choice is purely a function of our interests, and we make no claim that immersive VEs are better than other types of three-dimensional environments. We do, however, claim that immersive VEs are useful for certain tasks, domains, and applications because of their unique properties of immersion, immediacy, whole-body input, etc. Also, the general principles derived from this work should be applicable to many types of systems, and not only immersive VEs.

Fifth, we focus on single-user systems only. A large body of research into multi-user, collaborative VEs is emerging, and these have their own sets of issues related to interaction. Again, however, we feel that we must know more about the simple case in which only one user interacts with the environment before moving on to more complex multi-user VEs.

Finally, this work is restricted to a small number of physical input and output devices that are in common use. For display, all of our studies will use a head-mounted display (HMD), and simple, non-spatialized audio. We will not consider localized sound, haptics, olfactory feedback, or other non-standard forms of output. On the input side, we restrict our study to combinations of six degree of freedom trackers and simple button devices. No specialized input devices will be used or designed in this work. However, some of our experiments and applications will make use of passive physical props. These are non-instrumented physical objects that add realism, constraints, or other additional information to the virtual environment. For the most part, however, the techniques we discuss will differ only in their software implementation, not in the devices they use.

These decisions were not made arbitrarily. Rather, we are seeking to understand a simple subset of interaction techniques for VEs. This subset consists of techniques that can be implemented easily by anyone with a standard VE configuration. In many cases, it may be useful to go beyond these boundaries (for example, to build a new input device that matches a certain task), but the techniques we are studying are generally applicable to a wide range of possible applications.

1.5 Hypotheses

Our work covers a large territory in the overall field of VE interaction. However, there are three broad hypotheses that we have attempted to demonstrate in all phases of this research.

1. Intuition alone is not sufficient for the development of useful and usable (well-performing) interaction designs for VE applications.
2. Formal evaluation of VE interaction techniques will lead to specific and easily applied guidelines for the development of VE user interfaces.
3. The use of our formal methodology for the design and evaluation of VE interaction techniques will cause a measurable increase in the performance and usability of a real VE application to which evaluation results are applied.

We will refer to these hypotheses often throughout this thesis.

1.6 Contributions

This research makes a number of contributions to the fields of virtual environments, three-dimensional interaction, and HCI:

1. Our understanding of 3D interaction techniques has been extended from an intuitive feel for a technique's performance (often incorrect) to empirical measurements of performance and a formal understanding of the relationships between techniques.
2. The taxonomies and other parts of the design and evaluation framework provide a common ground for discussion and research in a more detailed and systematic fashion than simple lists of techniques or metaphors.
3. The combination of empirical results and formal frameworks provides the opportunity to create predictive models of technique performance.
4. The design and evaluation methodology can be reused to create and assess techniques for other VE interaction tasks.
5. The evaluation testbeds themselves can be reused to assess new interaction techniques for the tasks of travel, selection, and manipulation and compare their performance to previously tested techniques.
6. An indirect result of this research is a virtual environment application for environmental design education that has been shown to be both effective in its domain and to exhibit high levels of usability.
7. Finally, and perhaps most importantly, our experience in designing and evaluating VE interaction techniques has led to general principles and specific guidelines and recommendations (sections 1.8, 7.1) that can be used by application developers when creating highly interactive VEs.

1.7 Summary of This Work

In this chapter, we have introduced the subject of interaction techniques for VEs, motivated the need for research in this area, and defined the terms we will use, the scope of the work, our hypotheses, and our contributions.

Chapter two will present a detailed look at previous work that has influenced or informed the current research. This includes research into interaction in 2D interfaces, the evaluation of virtual environments, low-level perceptual and cognitive psychology work, and current three-dimensional user interfaces and interaction techniques.

Chapter three presents our design and evaluation methodology, with all of its component parts. This formal and systematic methodology is the abstract basis for the specific research that will be presented in later sections.

Chapter four applies this methodology to the task of travel, or user viewpoint movement control. We present descriptions of current travel techniques, taxonomies of techniques, and the results from five experiments comparing techniques for various tasks. We also discuss a travel testbed evaluation and its results.

In Chapter five, the methodology is applied to object selection and manipulation. Again, we discuss techniques from the literature, a taxonomy of techniques, and results of our evaluation of techniques. A testbed evaluation is also performed, and its results are presented in detail.

Chapter six describes a real-world VE application which is highly interactive. We discuss the initial two phases of interaction design for this application and the usability problems we encountered. We then describe the changes we made to the system based on the results of our evaluation, and the usability improvements that resulted.

Finally, we conclude in chapter seven with a discussion of the main contributions of this research and possibilities for future work in this area. In particular, this chapter contains detailed explanations of the guidelines and principles that have emerged from this research, so it will be of particular interest to application developers and interaction designers.

1.8 Summary of Recommendations

Our extensive design and evaluation of VE interaction techniques has led to a set of general principles and guidelines. Since these will likely be the most important legacy of this research, we list these recommendations here, and present a detailed exposition of them in chapter seven. The guidelines are divided into four categories: general principles for VE interaction, and guidelines for the design of travel, selection, and manipulation techniques.

1.8.1 Generic VE Interaction Guidelines

1. Do not assume that natural techniques will be the most intuitive or that they will have the best performance.
2. Provide redundant interaction techniques for a single task.

1.8.2 Guidelines for the Design of Travel Techniques

1. Make simple travel tasks simple by using target-specification techniques.
2. Avoid the use of teleportation; instead, provide smooth transitional motion between locations.
3. If steering techniques are used, train users in strategies to acquire survey knowledge. Use target-specification or route-planning techniques if spatial orientation is required but training is not possible.
4. Constrain the user’s travel to two dimensions if possible to reduce cognitive load.
5. Use non-head-coupled techniques for efficiency in relative motion tasks. If relative motion is not important, use gaze-directed steering to reduce cognitive load.

1.8.3 Guidelines for the Design of Selection Techniques

1. Use ray-casting techniques if speed of remote selection is a requirement.
2. Ensure that the chosen selection technique integrates well with the manipulation technique to be used.
3. If possible, design the environment to maximize the perceived size of objects.

1.8.4 Guidelines for the Design of Manipulation Techniques

1. Reduce the number of degrees of freedom to be manipulated if the application allows it.
2. Provide general or application-specific constraints or manipulation aids.

3. Allow direct manipulation with the virtual hand instead of using a tool.
4. Avoid repeated, frequent scaling of the user or environment.
5. Use indirect depth manipulation for increased efficiency and accuracy.