Interaction Techniques for Immersive Virtual Environments: Design, Evaluation, and Application

Doug A. Bowman
Graphics, Visualization, and Usability Center
College of Computing
Georgia Institute of Technology
bowman@cc.gatech.edu

ABSTRACT

Immersive virtual environments (VEs) have potential in many application areas, but most successful VE systems exhibit little interactivity. This is largely due to a lack of consideration or understanding of 3D interaction tasks and techniques. This work proposes the systematic study of design, evaluation, and application of VE interaction techniques. Design and evaluation are based on a formal task analysis and categorization of techniques, using multiple performance measures. This methodology will be tested by applying the results to a complex VE application allowing users to modify the design of a space while immersed within it.

1 INTRODUCTION

Virtual Environments (VEs) offer a new human-computer interaction paradigm in which users are no longer simply external observers of images on a computer screen but are active participants with a computer-generated three-dimensional virtual world. Proposed and developing applications include design, visualization, education, and both training and clinical uses in medicine. However, despite the rapid advances in the technology of displays, graphics processors, and tracking systems, and the advances in the realism and speed of computer graphics, there are still very few immersive VE applications in common use outside the research laboratory. This state of affairs is at least partly due to the lack of usable and effective interaction techniques (ITs) and user interface (UI) constructs for immersive VEs.

Therefore, in this work, we are designing effective and efficient new interaction techniques for VEs. However, we do not want to design haphazardly; rather, ITs will be designed in the context of a formal framework based on a task analysis and technique categorization. In addition, we are quantifying the performance of interaction techniques through experimental evaluation. Finally, we apply the results of the evaluation to real-world applications to verify its effectiveness.

How can we begin to analyze interaction techniques (ITs) 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. It is our claim in this work that most VE interactions fall into three task categories: *viewpoint motion control*, *selection*, and *manipulation*.

Viewpoint motion control, or travel, 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 which involves the picking of one or more virtual objects for some purpose. Manipulation refers to the positioning and orienting of virtual objects. 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). A fourth interaction task, 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), but at a low level, system control tasks can be characterized as selection and/or manipulation tasks.

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, interaction techniques for immersive VEs have been designed and developed in an ad hoc fashion, usually 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 design framework, or evaluated quantitatively against other techniques. Currently, then, we have a large collection of ITs for VEs, but we have no 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 novel techniques for each of the universal interaction 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.

2 METHODOLOGY

We wish to perform our design and evaluation of interaction techniques for immersive virtual environments in a principled, systematic fashion (see e.g. Price, Baecker, and Small, 1993, Plaisant, Carr, and Shneiderman, 1995). Formal frameworks provide us not only with a greater understanding of the advantages and disadvantages of current techniques, but also with better opportunities to create robust and well-performing new techniques, based on the knowledge gained through evaluation. Therefore, this research will follow several important design and evaluation concepts, elucidated in the following sections.

2.1 Taxonomization and Categorization

The first step in creating a formal framework for design and evaluation is to establish a *taxonomy* of interaction techniques for each of the universal interaction tasks. These taxonomies break up the tasks into separable components, each of which represents a decision that must be made by the designer of a technique. Some of these components are related directly to the task itself, while others may only be important as extensions of the metaphor on which the technique is based. In this sense, a taxonomy is the product of a careful task analysis.

Let us consider a simple example. Suppose the interaction task is to change the color of a virtual object (of course, this task could also be considered as a combination of universal interaction tasks: select an object, select a color, and give the “change color” command). A taxonomy for this task would include several task components. Selecting an object whose color is to change, choosing the color, and applying the color are components which are directly task-related. On the other hand, we might also include components such as the color model used or the feedback given to the user, which would not be applicable for this task in the physical world, but which are important considerations for an IT.

The taxonomies we establish for the universal tasks need to be correct, complete, and general. Any IT that can be conceived for the task should fit within the taxonomy, and should not contain components that are not addressed by the taxonomy. Thus, the components will necessarily be abstract. The taxonomy will also include several possible choices for each of the components, but we do not necessarily expect that each possible choice will be included. For example, in the object coloring task, a taxonomy might list touching the virtual object, giving a voice command, or choosing an item in a menu as choices for the color application component. However, this does not preclude a technique which applies the color by some other means, such as pointing at the object.

One way to verify the generality of the taxonomies we create is through the process of *categorization*. If existing techniques for the task fit well into the taxonomy, we can be more sure of its correctness and completeness. Categorization also serves as an aid to evaluation of techniques. Fitting techniques into a taxonomy makes explicit their fundamental differences, and we can determine the effect of choices in a more fine-grained manner. Returning to our example, we might perform an experiment comparing many different techniques for coloring virtual objects. Without categorization, the only conclusions we could draw would be that certain techniques were better than others. Using categorization, however, we might find that the choice of object selection techniques had little effect on performance, and that color application was the most important component in determining overall task time.

2.2 Guided Design

Taxonomization and categorization are good ways to understand the low-level makeup of ITs, and to formalize the differences between them, but once they are in place, they can also be used in the design process. We can think of a taxonomy not only as a characterization, but also as a design space. In other words, a taxonomy informs or guides the design of new ITs for the task, rather than relying on a sudden burst of insight.

Since a taxonomy breaks the task down into separable components, we can consider a wide range of designs quite quickly, simply by trying different combinations of choices for each of the components. There is no guarantee that a given combination will make sense as a complete interaction technique, but the systematic nature of the taxonomy makes it easy to generate designs and to reject inappropriate combinations.

Categorization may also lead to new design ideas. Placing existing techniques into a design space allows us to see the “holes” that are left behind – combinations of components that have not yet been attempted. One or more of the holes may contain a novel, useful technique for the task at hand. This process can be extremely useful when the number of components is small enough and the choices for each of the components are clear enough to allow a graphical representation of the design space, as this makes the untried designs quite clear (Card, Mackinlay, and Robertson, 1990).

2.3 Performance Measures

The overall goal of this research is to obtain information about human performance in common VE interaction tasks – but what is performance? As computer scientists, we tend to focus almost exclusively on speed, or time for task completion. Speed is easy to measure, is a quantitative determination, and is almost always the primary consideration when evaluating a new processor design, peripheral, or algorithm. Clearly, efficiency is important in the evaluation of ITs as well, but we feel there are also many other response variables to be considered.

Another performance measure that might be important is accuracy, which is similar to speed in that it is simple to measure and is quantitative. But in human-computer interaction, we also want to consider more abstract performance values, such as ease of use, ease of learning, and user comfort. For virtual environments in particular, presence might be a valuable measure. The choice of interaction technique could conceivably affect all of these, and they should not be discounted.

We should remember that the reason we wish to find good ITs is so that our applications will be more usable, and that VE applications have many different requirements. In many applications, speed and accuracy are not the main concerns, and therefore these should not always be the only response variables in our evaluations.

Also, more than any other computing paradigm, virtual environments involve the user – his senses and body – in the task. Thus, it is essential that we focus on user-centric performance measures. If an IT does not make good use of the skills of the human being, or if it causes fatigue or discomfort, it will not provide overall usability despite its performance in other areas. In this work, then, we will evaluate based on multiple performance measures that cover a wide range of application and user requirements.

2.4 Testbed Evaluation

To evaluate ITs, we could perform any of a number of possible evaluation techniques, including usability studies, cognitive walkthroughs, or formal experiments. These experimental methods and other evaluation tools can be quite useful for gaining an initial understanding of interaction tasks and techniques, and for measuring the performance of various techniques in specific interaction scenarios. However, there are some problems associated with using these types of tests alone.

First, while results from informal evaluations can be enlightening, they do not involve any quantitative information about the performance of interaction techniques. Without statistical analysis, key features or problems in a technique may not be seen. Performance may also be dependent on the application or other implementation issues when usability studies are performed.

On the other hand, formal experimentation usually focuses very tightly on specific technique components and aspects of the interaction task. An experiment may give us the information that technique X performs better than technique Y in situation Z, but it is often difficult to generalize to a more meaningful result. Techniques are not tested fully on all relevant aspects of an interaction task, and generally only one or two performance measures are used.

Finally, in most cases, traditional evaluation takes place only once and cannot truly be recreated later. Thus, when new techniques are proposed, it is difficult to compare their performance against those that have already been tested.

Therefore, we propose the use of *testbed evaluation* as the final stage in our analysis of interaction techniques for universal VE interaction tasks. This method addresses the issues discussed above through the creation of testbeds – environments and tasks that involve all of the important aspects of a task, that test each component of a technique, that consider outside influences (factors other than the interaction technique) on performance, and that have multiple performance measures.

As an example, consider a proving ground for automobiles. In this special environment, cars are tested in cornering, braking, acceleration, and other tasks, over multiple types of terrain, and in various weather conditions. Task completion time is not the only performance variable considered. Rather, many quantitative and qualitative results are tabulated, such as accuracy, distance, passenger comfort, and the "feel" of the steering.

The VEPAB project (Lampton et al, 1994) was one research effort aimed at producing a testbed for VEs, including techniques for viewpoint motion control. It included several travel tasks that could be used to compare techniques. However, this testbed was not based on a formal understanding of the tasks or techniques involved.

In this work, we will create a series of testbeds for the universal VE interaction tasks of viewpoint motion control, selection and manipulation, and system control. Together, these testbeds make up VR-SUITE – the Virtual Reality Standard User Interaction Testbed Environment. The testbeds will allow us to analyze many different ITs in a wide range of situations, and with multiple performance measures. Testbeds will also be based on the formalized task and technique framework discussed earlier, so that the results will be more generalizable. Finally, the environments and tasks will be standardized, so that new techniques can be run through the appropriate testbed, given scores, and compared with other techniques that were previously tested.

3 COMMON INTERACTION TASKS

3.1 Viewpoint Motion Control

Our first studies (Bowman, Koller, and Hodges, 1997) were aimed at analysis and evaluation of techniques for the most ubiquitous VE interaction: travel. A travel technique simply refers to the mechanism used to move one's viewpoint between different locations in a virtual environment. Travel is part of the larger task of navigation, which includes both the actual movement and the decision process involved in determining the desired direction and target of travel (wayfinding).

Our analysis of this task identified three basic components that must be included in any travel technique: *direction/target selection* (the means by which the user indicates the direction of motion or the endpoint of the motion), *velocity/acceleration selection* (the means by which the user indicates the speed and acceleration of the motion), and *conditions of input* (the means by which the user begins, continues, and ends the motion). These three components provide the organizational structure for a preliminary taxonomy of travel techniques (Figure 1).

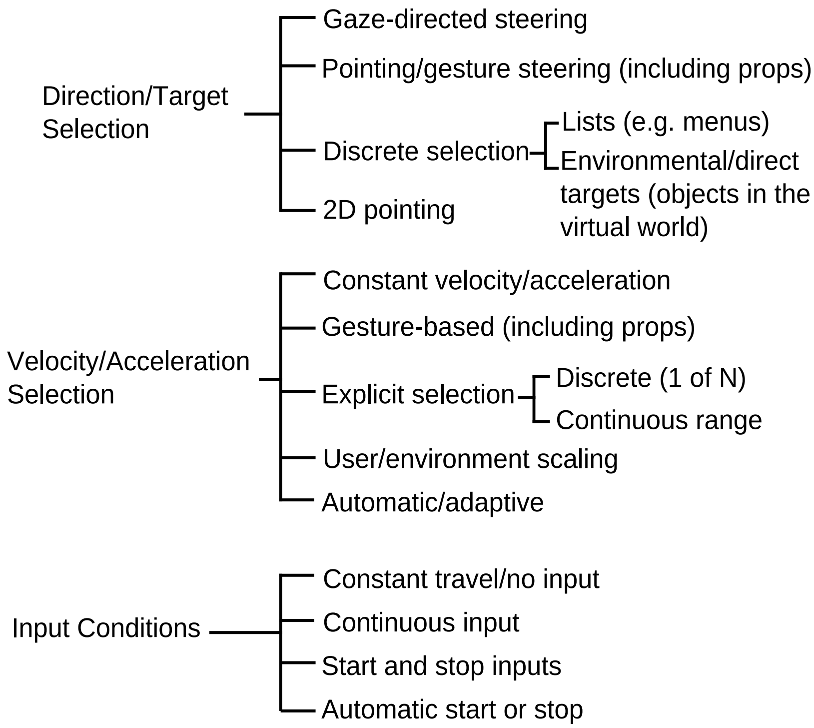

Figure 1. Preliminary taxonomy of immersive VE travel techniques

Our research also identified a set of *quality factors*, or performance metrics, by which we could evaluate travel techniques. These include quantitative measures such as speed and accuracy, HCI concerns such as ease of use and ease of learning, and more subjective metrics such as spatial awareness, presence, and user comfort. Our evaluation philosophy was to compare technique components from the taxonomy on the basis of these quality factors, without reference to any specific applications. In this way, application developers could specify desired levels of performance for any or all of the quality factors, and choose technique components that had been shown to fit those requirements.

We performed three initial experiments based on this philosophy. The first two experiments compared a pair of very common direction selection techniques: gaze-directed steering (the user looks in the desired direction of travel) and pointing (the user points his hand in the desired direction of travel) (Mine, 1995). The evaluation was performed on the basis of speed and accuracy. We found that there was no significant difference between the techniques for a simple, straight-line motion with a visible target destination, but that the pointing technique performed significantly better ($p < 0.025$) in a *relative motion* task (that is, travel where the target is not explicit, but instead is defined relative to the position and orientation of some object in the environment). This task gets at the heart of the difference between the two techniques: gaze-directed steering forces the user to look in the direction of motion while pointing allows the user to look in one direction and move in another.

The third experiment compared various velocity and acceleration techniques on the basis of spatial awareness. We hypothesized that users would be more or less aware of their surrounding environment after travel depending on the speeds and accelerations they had experienced during motion. We found that users were significantly more disoriented ($p < 0.01$) after the use of a “jumping” technique (where users are instantly transported to the target destination) than after using any of 3 other continuous motion techniques.

Our initial investigations led us to realize that performance differences could be influenced by a wide variety of factors other than the interaction technique. In our latest work, we describe an expanded evaluation framework, which explicitly includes outside factors in the model of performance. Outside factors include task characteristics (e.g. distance to travel, number of turns in the path), environment characteristics (e.g. number of obstacles, level of visual detail), system characteristics (e.g. rendering style, frame rate), and user characteristics (e.g. length of reach, experience with VE technology). We also performed a fourth experiment incorporating this expanded framework. In it, we compared three direction selection techniques on the amount of cognitive load they placed on the user. Our findings support the use of the enlarged framework: technique was not a significant factor, but the dimensionality of the environment (1-, 2-, or 3-dimensional paths were used) was significant ($p < 0.01$).

Based on these experiences and observations of VE travel techniques, we are currently in the process of reworking the taxonomy and designing tasks and environments that will be part of a viewpoint motion control testbed.

3.2 Selection and Manipulation

We have also begun an initial investigation into interaction techniques for selection and manipulation of virtual objects. Selection refers to the act of specifying or choosing an object for some purpose. Manipulation is the task of setting the position and orientation (and possibly other characteristics such as shape) of a selected object. Manipulation requires a selection technique, but the opposite is not always true. Selection techniques can be used alone for tasks such as choosing a menu item or deleting an object.

The most obvious and common set of techniques for these interactions is the real-world metaphor of in-hand manipulation. The user selects an object by “touching” it with his virtual hand, and manipulates it directly by moving his hand. This is intuitive and cognitively simple, but has limited practicality. Many virtual objects are too large to allow easy placement while close enough to touch the object. Also, it is inappropriate to force the user to move within arm’s reach of an object to manipulate it, especially if the application requires multiple manipulations and efficient performance. Therefore, we are mainly interested in techniques that allow selection and manipulation at a distance.

To begin to understand the tasks involved and the set of published techniques, we conducted an informal user study comparing several of the ITs (Bowman and Hodges, 1997). Two basic categories of techniques were represented: ray-casting and arm-extension. In a ray-casting technique (Mine, 1995), a light ray emanates from the user’s virtual hand. To select an object, the user intersects the object with the light ray and performs a “grab” action (usually by pressing a button). She can then manipulate the object using the light ray. Arm-extension techniques (e.g. Poupyrev et al, 1996) allow the user to reach faraway objects by providing a means to make the virtual arm longer than the user’s physical arm. This can be accomplished by various mapping strategies, button presses, etc. The user then selects and manipulates the object as with the in-hand metaphor: touch the object with the virtual hand and manipulate it with hand movements.

We found that none of the tested techniques provided optimal usability or usefulness, but instead all involved tradeoffs. In general, ray-casting techniques proved best for object selection, but arm-extension techniques allowed more precise and expressive object manipulation. Based on this observation, we developed the HOMER (Hand-centered Object Manipulation Extending Ray-casting) technique, which combines the two metaphors seamlessly to allow ease of selection and manipulation for objects at any distance. The user selects an object by intersecting a light ray with it, and when the selection is made, the user’s virtual hand extends so that it touches the selected object. The object can then be manipulated directly with the virtual hand, until it is released, at which point the virtual hand returns to its normal position.

We are currently in the beginning stages of the development of a formalized evaluation framework for these tasks similar to the one for the travel interaction described above. We have identified initial sets of task components, technique categories, performance metrics, and outside factors which could influence performance. A preliminary implementation of a testbed for selection and manipulation has also been developed. Each trial (see Figure 2) requires the user to select the center object from a group of objects and place it within a transparent target. We vary the size of the object, density of the group, distance to the object, size of the target, distance to the target, and number of degrees of freedom the user must control.

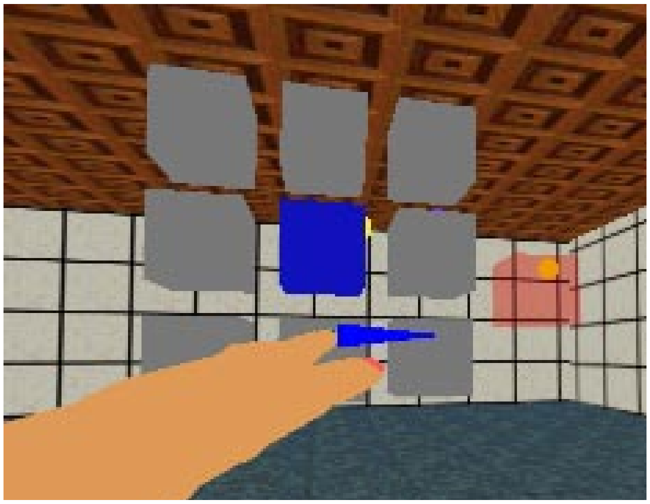

Figure 2. Example Trial Setup in the Selection/Manipulation Testbed.

4 APPLICATION

Our testbeds should produce important results regarding the performance of various ITs for travel, selection, and manipulation. However, we must keep in mind that the ultimate goal of such research is to produce useful and usable VE systems for real-world applications. Therefore, we have been applying the results of our work to an interesting and complex VE application: *immersive design*. One of the most popular VE applications is the architectural walkthrough (Brooks et al, 1992), which allows real-time viewing of an architectural space, but no opportunities to modify that space. In an immersive design system, users can create or modify a 3-dimensional space while immersed within it. This is an extreme departure from traditional design paradigms, but has the potential to tighten the design cycle and to allow designers immediate and realistic feedback on the visual impact of their creations.

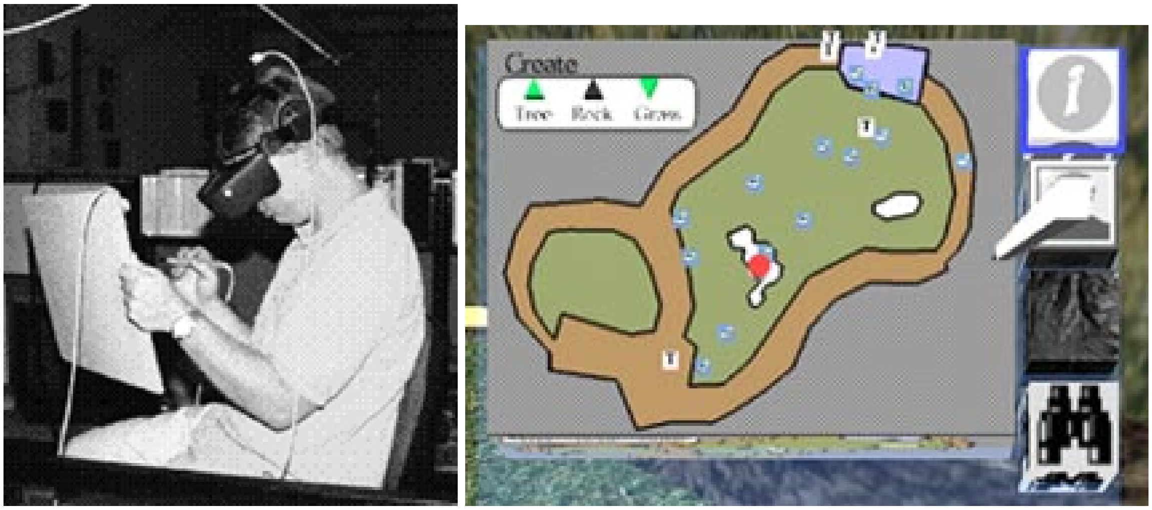

Figure 3. Physical (left) and virtual (right) views of the pen & tablet interaction metaphor

Our latest design application is built on top of the VR Gorilla Exhibit (Allison et al, 1997). In this application, we focused not on the conceptual stages of design, but instead on the detailed design of domain-specific elements. Using the system, designers can make changes to the design of a pre-existing zoo exhibit, including the terrain, visitor viewpoints, and visual elements such as trees and rocks.

Two interaction metaphors are combined to allow these design changes to be made in an efficient and usable manner. First, travel, selection, and manipulation can all be performed directly in the 3D environment. Users can point in the direction they wish to move and can use an arm-extension technique to grab objects such as trees and move them around. All of these interactions are well constrained so that the user is not overwhelmed. Second, the tasks can be done indirectly using a "pen & tablet" metaphor (Angus & Sowizral, 1995). Here, the user holds a physical tablet and stylus, both of which are tracked (Figure 3, left). In the VE, a 2D user interface is seen on the tablet surface, and the stylus can be used to press buttons or drag icons on this interface (Figure 3, right). This application was recently used by students in a class on environmental design, who found it easy to learn and use, and who produced a number of unique and practical designs after only a brief session with the system.

ACKNOWLEDGMENTS

The author wishes to thank his advisor, Dr. Larry F. Hodges, and those who have worked on various aspects of this work: David Koller, Don Allison, Brian Wills, and Jean Wineman.

REFERENCES

Allison, D., Wills, B., Bowman, D., Wineman, J., and Hodges, L. (1997). The Virtual Reality Gorilla Exhibit. *IEEE Computer Graphics & Applications*, *17*(6), November/December, 30-38.

Angus, I. and Sowizral, H. (1995). Embedding the 2D Interaction Metaphor in a Real 3D Virtual Environment. *Proceedings of SPIE, Stereoscopic Displays and Virtual Reality Systems*, *2409*, 282-293.

Bowman, D. and Hodges, L. (1997). An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. *Proceedings of the ACM Symposium on Interactive 3D Graphics*, 35-38.

Bowman, D., Koller, D., and Hodges, L. (1997). Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. *Proceedings of Virtual Reality Annual International Symposium*, 45-52.

Brooks, F. et al, (1992). Final Technical Report: Walkthrough Project. Report to National Science Foundation.

Card, S., Mackinlay, J., and Robertson, G. (1990). The Design Space of Input Devices. *Proceedings of CHI*, 117-124.

Foley, J. (1979). A Standard Computer Graphics Subroutine Package. *Computers and Structures*, *10*, 141-147.

Lampton, D., Knerr, B., Goldberg, S., Bliss, J., Moshell, J., and Blau, B. (1994). The Virtual Environment Performance Assessment Battery (VEPAB): Development and Evaluation. *Presence: Teleoperators and Virtual Environments*, *3*(2), 145-157.

Mine, M., (1995). Virtual Environment Interaction Techniques. UNC Chapel Hill Computer Science Technical Report TR95-018.

Plaisant, C., Carr, D., and Shneiderman, B. (1995). Image-Browser Taxonomy and Guidelines for Designers. *IEEE Software*, March.

Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. (1996). The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 79-80.

Price, B., Baecker, R., and Small, I. (1993). A Principled Taxonomy of Software Visualization. Journal of Visual Languages and Computing.