Journal of Visual Languages and Computing (1999) 10, 37–53
Article No. jvlc.1998.0111, available online at http://www.idealibrary.com on IDEAL

Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments

DOUG A. BOWMAN AND LARRY F. HODGES

Graphics, Visualization, and Usability Center, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, U.S.A., e-mail: {bowman, hodges}@cc. gatech.edu

Immersive virtual environments (VEs) have potential in many application areas, but many complex VE systems exhibit usability and interaction problems. This is partly due to a lack of consideration or understanding of 3D interaction tasks and techniques. This paper proposes the systematic study of the design, evaluation, and application of VE interaction techniques. In this methodology, design and evaluation are based on a formal task analysis and categorization of techniques, using multiple performance measures. As a direct consequence of our use of this methodology, we also present a variety of novel designs and evaluation results with respect to interaction techniques for three common VE tasks.

1. Introduction

IMMERSIVE 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 (3D) 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 in the realism and speed of computer graphics, there are still very few immersive VE applications in common use outside the research laboratory. Such applications include architectural walkthrough [1] (and other passive visualizations), phobia therapy [2], and entertainment. It is instructive to note the similarities among these three categories of applications: none of them require complex interaction between the user and the system. Head tracking and some method of navigating the 3D space are usually sufficient. Although the user may be interacting frequently, the interactions are mostly simple in nature.

One can conclude from this analysis that interaction is a major reason for the lack of real-world usage of more complex VE applications. There seems to be, in general, little understanding of human–computer interaction (HCI) in three dimensions, and a lack of knowledge regarding the effectiveness of interaction in VEs, although some recent work has begun to address these issues. Many researchers hold to the intuitive notion that

1045-926X/99/010037+17 $30.00/0

interaction in VEs should replicate our interaction with the physical world (see e.g. Nielsen [3]), but such interaction is never completely realistic, and severely limits the potential for productivity. We claim, instead, that VEs should enhance the physical, cognitive, and perceptual capabilities of the user, allowing them to do things that are impossible in the real world.

Therefore, this paper will describe a methodology with which we can improve the usability of interactively complex VE applications through careful attention to interaction techniques (ITs). Specifically, we will focus on the design, evaluation, and application of ITs for such systems.

How can we begin to analyze ITs for immersive virtual environments? There are a multitude of tasks (desired user actions) which one might conceivably want to perform within a VE, and many of them are domain- or 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 [4].

By identifying and studying these basic interaction tasks, we are better equipped to design and evaluate ITs that perform well for those tasks. Since these tasks are so common, the resulting ITs may be useful in a wide range of VE applications. Thus, the identification and understanding of such 'universal tasks' for VEs is an important first step towards addressing usability difficulties within interactively complex VE applications. Many 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 most immersive VEs use head tracking to control 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 generally be characterized as selection and/or manipulation tasks. Since system control can be considered a compound task, we do not design or evaluate system control techniques explicitly.

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 (e.g. the available hardware, the experience of the user, or the precision required by the task).

Many 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 a few notable exceptions, ITs were not designed with regard to any explicit design framework, or evaluated quantitatively against other techniques. Currently, then, we have a collection of ITs for VEs, but little in-depth understanding of their characteristics or analysis of their relative performance.

This paper will describe a research methodology for the design, evaluation, and application of interaction techniques for immersive VEs, as well as early results of the use of the methodology. These methods emerged gradually as a result of our extensive experience in designing VE applications with complex interactivity, evaluating existing interaction techniques, and designing novel techniques. Keeping in mind that the ultimate goal of such research is to increase the usefulness and usability of complex VE applications, we will discuss eight steps toward the accomplishment of this goal:

1. To perform initial evaluations of tasks and techniques in order to gain insight into their fundamental nature.
2. To develop formal characterizations of the universal interaction tasks and formal taxonomies and categorizations of interaction techniques for those tasks.
3. To use these characterizations to design novel techniques for each of the universal interaction tasks.
4. To list factors other than interaction techniques that might have a significant effect on task performance.
5. To list multiple performance metrics for VE interaction tasks along with methods for measuring performance in each area.
6. To develop and utilize quantitative and general experimental analyses for the purpose of comparing the performance of interaction techniques for the universal tasks.
7. To gather results of these analyses into a database for the purposes of modeling human performance on VE interaction tasks.
8. To apply the experimental results to VE applications by choosing interaction techniques for those applications which meet their specified interaction requirements.

2. Methodology

Principled, systematic design and evaluation frameworks (see e.g. Price *et al.* [5] and Plaisant *et al.* [6]) give formalism and structure to research on interaction, rather than relying solely on experience and intuition. 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, we follow several important design and evaluation concepts, elucidated in the following sections. Figure 1 presents an overview of this methodology, and will be referred to frequently in the following sections.

2.1. Initial Evaluation

The first step towards formalizing the design, evaluation, and application of interaction techniques is to gain an intuitive understanding of the tasks and current techniques

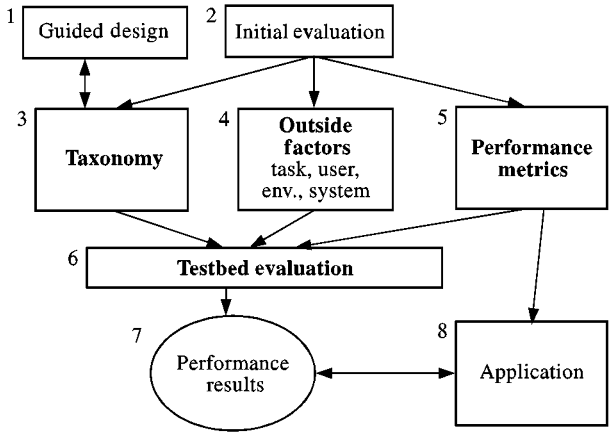

Figure 1. Design, evaluation, and application methodology

available for the tasks (Figure 1, #2). This is accomplished through experience using ITs and through observation and evaluation of groups of users. Often in this phase, we perform informal user studies or usability tests, asking users what they think of a particular technique, or observing them trying to complete a given task with the technique. These initial evaluation experiences are drawn upon heavily for the processes of taxonomization and categorization (Section 2.2), listing outside influences on performance (Section 2.4), and listing performance measures (Section 2.5). It is helpful, therefore, to gain as much experience of this type as possible so that good decisions can be made in the next phases of formalization.

2.2. Taxonomization and Categorization

Our next step in creating a formal framework for design and evaluation is to establish a *taxonomy* (Figure 1, #3) of interaction techniques for each of the interaction tasks described above. Figure 2 shows a generalized taxonomy. Such taxonomies decompose the tasks into separable subtasks, each of which represents a decision that must be made by the designer of a technique. Some of these subtasks 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. Once the task has been broken up to a sufficiently fine-grained level, the taxonomy is completed by listing possible methods (technique components) for accomplishing each of the lowest-level subtasks. An interaction technique is made up of one technique component from each of the lowest-level subtasks, such as the set of shaded components in Figure 2.

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 subtasks. Selecting an object whose color is to change, choosing the color, and applying the color are subtasks that are directly task-related. On the other hand, there are IT components that are not

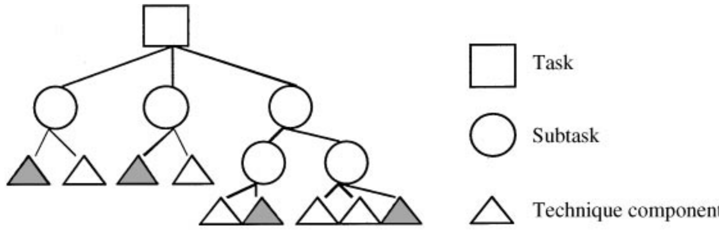

Figure 2. General taxonomy format

directly related to the user task, such as the color model used or feedback given by the system. These items need to be accounted for in a complete taxonomy of ITs, even though they are not strictly user tasks, so they are included as system characteristics (see Section 2.4).

Ideally, 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. Thus, the subtasks will necessarily be abstract. The taxonomy will also list several possible technique components for each of the subtasks, but we do not claim to list each conceivable component. 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 subtask. 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*—classifying existing ITs within the framework of the taxonomy. If existing techniques for the task fit well into a 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 the choice of the color application technique component was most important component in determining overall task time.

2.3. 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 (Figure 1, #1), rather than relying on a sudden burst of insight.

Since a taxonomy breaks the task down into separable subtasks, we can consider a wide range of designs quite quickly, simply by trying different combinations of technique components for each of the subtasks. 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 subtasks is small enough and the choices for each of the subtasks are clear enough to allow a graphical representation of the design space, as this makes the untried designs quite obvious [7].

2.4. Outside Influences on Performance

Interaction techniques cannot be evaluated in a vacuum. A user's performance on an interaction task may depend on a variety of factors (Figure 1, #4), of which the interaction technique is but one. In order for our design and evaluation framework to be complete, we must include such factors explicitly, and use them as secondary independent variables in our evaluations. We have identified four categories of outside factors for our methodology.

First, *task characteristics* are those attributes of the task that may affect performance, including the distance to be traveled or the size of the object being manipulated. Second, we consider *environment characteristics*, such as the number of obstacles and the level of activity or motion in the VE. *User characteristics*, including cognitive measures such as spatial ability or physical attributes such as arm length, may also contribute to performance. Finally, *system characteristics* may be significant, such as the lighting model used or the mean frame rate.

2.5. Performance Measures

Our methodology is designed to obtain information about human performance in common VE interaction tasks—but what is performance? Speed, or task completion time, is easy to measure, is a quantitative determination, and is clearly important in the evaluation of ITs, but we feel there are also many other performance metrics (Figure 1, #5) 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 (HCI), 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 [8] might be a valuable measure. These values refer to the performance of the *technique*, rather than human performance, but are important nonetheless. 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 that our applications will be more usable, and that VE applications have many different requirements. In some applications, speed and accuracy may not be 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 [9]. 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. Our methodology, then, recommends evaluation based on multiple performance measures that cover a wide range of application and user requirements.

2.6. Testbed Evaluation

To evaluate ITs, there are many available 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 use *testbed evaluation* (Figure 1, #6) 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.

Testbeds are a common evaluation method in engineering, but have not been widely used in HCI for interface or interaction technique evaluation. The VEPAB project [10] 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. VRMAT [9] is a testbed environment aimed at evaluation of 3D manipulation techniques. It has a somewhat different focus than our testbeds, as it attempts to provide a generalized framework within which specific experiments can be implemented, without making judgments about the validity of those experiments. Our testbeds, on the other hand, define a specific set of evaluations that theoretically test each important aspect of a technique for a given interaction task.

We have implemented a testbed for the tasks of selection and manipulation, and are in the process of designing a similar environment in which to evaluate techniques for travel. The testbeds allow us to analyze many different ITs in a wide range of situations, and with multiple performance measures. Testbeds are also based on the formalized task and technique framework discussed above, so that the results are more generalizable. Finally, the environments and tasks are standardized, so that new techniques can be run through the appropriate testbed, given scores, and compared with other techniques that were previously tested.

2.7. Performance Results

Testbed evaluation produces a set of results or models (Figure 1, #7) that characterize the performance of an interaction technique for the specified task. Performance is given in terms of multiple performance metrics, with respect to various levels of outside factors. These results become part of a performance database for the interaction task, with more information being added to the database each time a new technique is run through the testbed.

2.8. Application of Results

The last step in our methodology is to apply the performance results to VE applications (Figure 1, #8), with the goal of making them more useful and usable. In order to choose interaction techniques for applications appropriately, we must understand the interaction requirements of the application. We cannot simply declare one best technique, because the technique that is best for one application will not be optimal for another application with different requirements. For example, a VE training system will require a travel technique that maximizes the user's spatial awareness, but this application will not require a travel technique that maximizes point-to-point speed. On the other hand, in a battle planning system, speed of travel may be the most important requirement.

Therefore, applications need to specify their interaction requirements before the correct ITs can be chosen. This specification will be done in terms of the performance metrics which we have already defined as part of our formal framework. Once the requirements are in place, we can use the performance results from testbed evaluation to recommend ITs that meet those requirements.

3. Evaluation Results

3.1. Viewpoint Motion Control

In our first studies [11], we focused on the analysis and evaluation of techniques for the most ubiquitous VE interaction: travel (also called viewpoint motion control). Travel simply refers to the movement of 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).

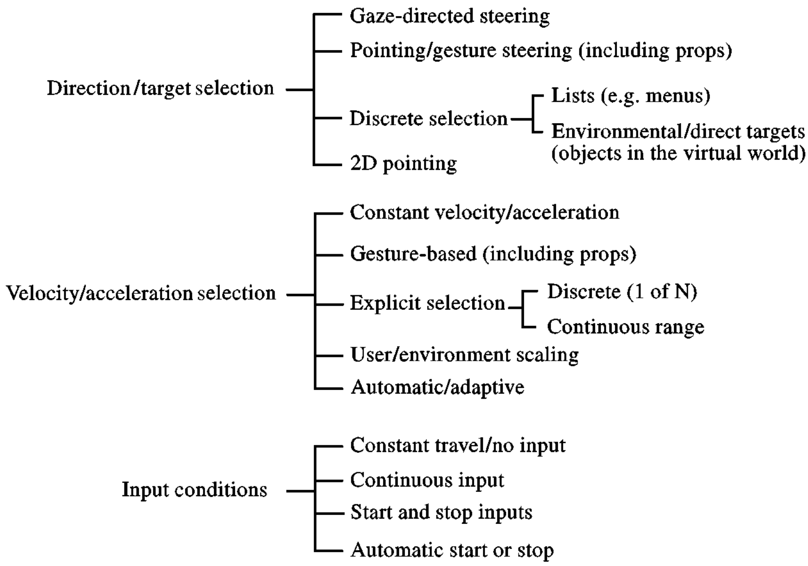

Figure 3. Taxonomy of immersive VE travel techniques

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 subtasks provide the organizational structure for a taxonomy of travel techniques (Figure 3).

Our research also identified a set of 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. As stated above, our evaluation philosophy is to compare technique components from the taxonomy on the basis of these performance metrics, without reference to any specific applications. In this way, application developers can specify desired levels of performance for any or all of the performance metrics, and choose technique components that have been shown to fit those requirements.

We performed three initial experiments based on this philosophy. All three experiments used a within-subjects design with travel technique as the main independent variable and task completion time as the response variable. (For full details on these experiments, we refer the reader to Bowman *et al.* [11].) 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) [12]. Eight undergraduate subjects participated in the study, with each subject completing 640 trials. 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 our nine subjects were significantly more disoriented ($p < 0.01$) after the use of a 'jumping' technique (where users are instantly teleported to the target destination) than after using any of three 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 [13], 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 (for details, see Bowman *et al.* [13]) incorporating this expanded framework. In it, we compared three direction selection techniques (gaze, pointing, and torso-directed) on the amount of information users could gather while traveling. Twenty-six subjects traveled along paths and attempted to gather as much information as possible while also traveling as quickly as possible. A within-subjects design was used, where travel technique, path dimension, and the use of collision detection were independent variables, and the users' scores on a test of their memory of the path was the dependent variable. The interaction technique used did not prove to be significant, because the most cognitively difficult techniques also support better information gathering, and *vice versa*. However, our findings support the use of the enlarged framework: the dimensionality of the environment (1D, 2D, or 3D paths were used) was a significant factor ($p < 0.01$).

3.2. Selection and Manipulation

We have also investigated 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 scale or shape) of a selected object. Manipulation requires a selection technique, but the converse 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.

In order to understand the tasks involved and the set of published techniques, we conducted an informal user study with 11 student volunteers comparing several of the ITs [14]. Two basic categories of techniques were represented: ray-casting and arm-extension. In a ray-casting technique [12], 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 (see e.g. Poupyrev *et al.* [15]) 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. Users praised arm-extension techniques for their ease of object manipulation, but disliked the imprecise selection and finite reach characteristic of some of these techniques. On the other hand, all subjects found manipulation difficult with ray-casting techniques because the object is attached to the light ray instead of directly to the hand, but subjects reported that selection was much easier with ray-casting. 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.

Since this initial evaluation and design, we have produced a formal design and evaluation framework for selection and manipulation techniques. Figure 4 shows our current selection and manipulation taxonomy, where the entire task is broken down initially into three subtasks: selection, manipulation, and release. These subtasks are further divided; for example, manipulation involves attachment, positioning, and orientation.

This taxonomy has already been used extensively for guided design of selection and manipulation techniques. We have implemented software modules for five important subtasks from the taxonomy: indication of object (eight technique components), attachment (6), positioning (6), orientation (4), and release (4). We can combine the technique components in any meaningful way to produce a complete IT. There are 4608 possibilities overall, but this number has been reduced to around 600 techniques through dependencies and constraints. The capability to combine components in new ways with the touch of a button has already resulted in several interesting techniques, including a HOMER-like technique that uses gaze-based selection rather than pointing, and a family of techniques that specify positioning and orientation with two separate hand trackers rather than a single tracker for both.

As we did for travel, we have also created lists of performance metrics and outside factors that could be important for selection and manipulation interactions. The

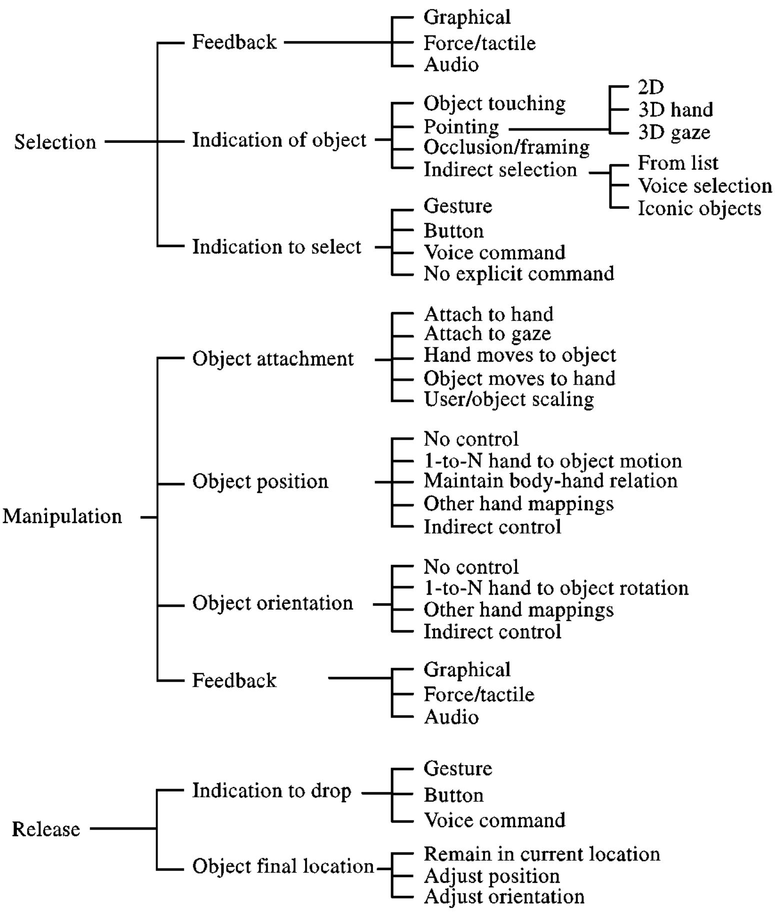

Figure 4. Selection/manipulation taxonomy

performance measures are similar to those for travel, but include some metrics specific to these interactions, such as expressibility (the expressive power of the technique to specify a range of object positions and orientations). Again, the outside factors include task characteristics (e.g. number of degrees of freedom to be manipulated), environment characteristics (e.g. density of objects in the scene), user characteristics (e.g. arm length), and system characteristics (e.g. stereo vs. mono viewing).

A testbed for selection and manipulation techniques has also been developed. This testbed provides a range of tasks and environments such that interaction techniques are fully tested. The experiment measures multiple performance variables, such as speed, accuracy, and user comfort (via subjective reports). Each trial (Figure 5) requires the user to select the center object from a group of objects and place it within a target area. We vary the size of the objects, density of the group, and distance to the objects as selection variables, and the size of the target, distance to the target, and number of degrees of freedom the user must control as manipulation variables.

Currently, we have evaluated nine techniques, using five subjects each, within the testbed. We first evaluated the Go-Go technique [15], a well-known arm-extension

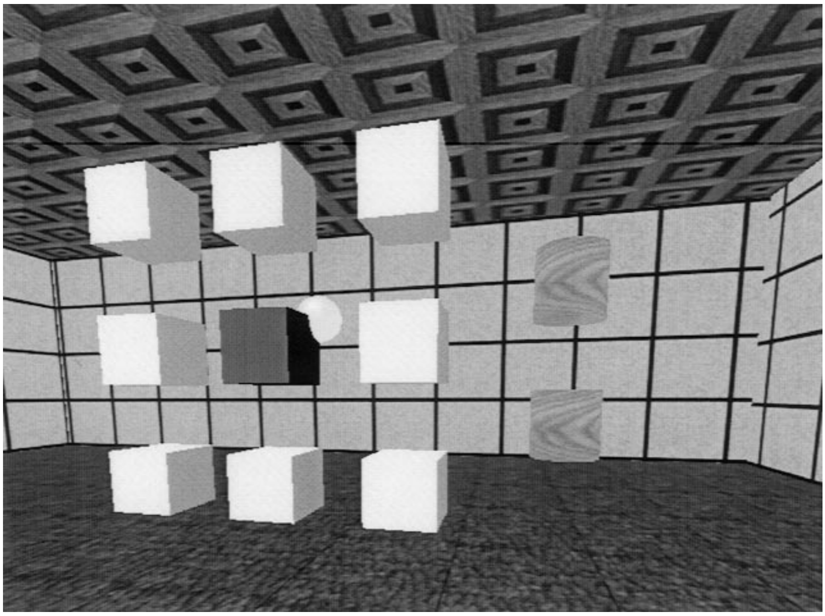

Figure 5. Example trial setup in the selection/manipulation testbed

technique. We also evaluated eight techniques based on three subtasks from the selection and manipulation taxonomy. Two selection techniques were considered: ray-casting and occlusion selection. We also considered two methods of attachment: moving the user's hand to the selected object (as in the HOMER technique), and scaling the user so the virtual hand touches the selected object. Finally, we considered two techniques for moving objects: a direct mapping from hand motion to object motion, and the use of buttons to move objects closer to or farther away from the user. Each user completed 48 trials, half of which measured selection performance, and half of which measured manipulation performance.

Our initial analysis of the results of this evaluation indicates that the variables we chose to manipulate in the testbed can indeed significantly influence the performance of selection and manipulation techniques. We also found some significant differences between techniques that will inform the interaction design of our environmental design application (see below).

We found a significant difference between the mean selection time for the Go-Go technique and the other two selection techniques ($p < 0.001$), indicating that while the Go-Go technique may be more natural due to its consistent virtual hand metaphor, speed and ease of selection are enhanced with the ray-casting or occlusion selection methods. For manipulation, we obtained the surprising result that techniques using joystick buttons to move the selected object in and out were significantly faster than their counterparts using hand motion only ($p < 0.001$). Again, the use of buttons may seem awkward and unnatural, but it can improve manipulation speed, especially when placement is required to be extremely accurate.

We also found that almost all of the task and environment variables were significant factors in performance. For selection, distance and size of the object were both significant, while distance to the target, size of the target, and the number of required degrees of freedom all had significant effects on manipulation performance. Finally, our use of multiple performance measures was validated, as we found significant correlation between selection technique and reported arm strain, attachment technique and reported dizziness (the user scaling technique causes dizziness in some users), and manipulation technique and reported arm strain. Complete analysis of the results of our testbed evaluation of selection and manipulation technique will be reported in a later article.

3.3. Example 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 [1], 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 three-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.

We have previously developed and evaluated an immersive design system called the conceptual design space (CDS) [16]. This system was used by architecture students to help them complete a design project. Users could load in pre-built CAD models, or build from scratch while immersed. Once a conceptual model was in place, users could visualize it in real-time and make modifications to object position, scale, color, and texture. The system suffered, however, from some difficult usability problems, which motivated our current work.

Our latest design application, the Virtual Habitat [17], is built on top of the VR Gorilla Exhibit [18]. In this application, we are focusing 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.

Interaction in the Virtual Habitat was based in part on our original evaluation of travel, selection, and manipulation techniques. Two interaction metaphors are combined to allow 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 [19]. Here, the user holds a physical tablet and stylus, both of which are tracked (Figure 6, 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 6, right).

This application was recently used by 24 students (eight teams of three students each) in a class on environmental design, who found the Virtual Habitat relatively easy to learn

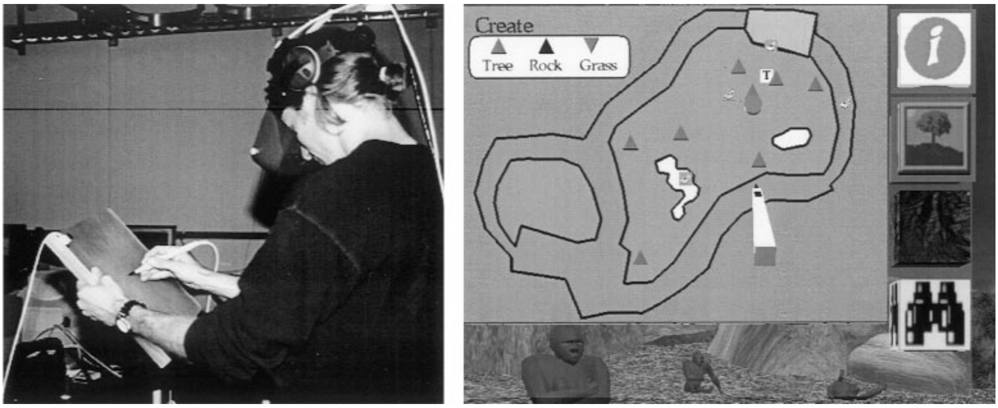

Figure 6. Physical (left) and virtual (right) views of the pen & tablet interaction metaphor

and use, and who produced a number of unique and practical designs after only a brief session with the system. As with CDS, a usability study was performed on the Virtual Habitat to quantify its performance and ease of use. We asked students to rate the usability of various types of interactions on a scale of 1–5, with 5 representing the most usable. The overall average of these responses was 3.96. Interestingly, we found that interaction using the pen & tablet metaphor received much higher ratings than direct manipulation in the virtual world. For example, movement accomplished by dragging the user icon on the tablet rated 4.21, while direct movement using the pointing technique rated only 3.71. Two potential conclusions can be drawn from this. First, we could conclude that accessibility and constraints are two important factors affecting interaction performance in immersive environments. The tablet is always accessible to the user, the icons are never out of reach, and the physical surface provides an important constraint; therefore these techniques exhibit higher performance. Second, we could conclude that user familiarity is the key to satisfaction. The pen & tablet metaphor more closely mimics 2D interaction on the desktop, so users are more comfortable with this mode of interaction. Both of these conclusions imply that extra care must be taken in the design of 3D direct manipulation interaction techniques.

When our testbed evaluation is complete, we will develop and implement a new interaction design for the Virtual Habitat based on our systematic methodology. We have created sets of interaction requirements for this application in terms of the performance metrics, and will choose ITs that meet those requirements for this final design. In order to verify the usefulness of our design and evaluation methods, we will perform another usability study on the system to show that the efficiency, effectiveness, and usability of the application has improved.

4. Conclusions

In this paper, we have proposed a formal, systematic methodology for the design, evaluation, and application of interaction techniques for immersive virtual environments. Work in this area is needed because of the lack of understanding of three-dimensional and immersive interaction, and because usability and interaction difficulties have slowed the development of promising potential VE applications. Our methodology provides a deeper understanding of VE interaction tasks, a systematic method for designing new techniques, and evaluation testbeds that can be used to compare techniques along many different dimensions. It also includes methods for choosing appropriate ITs for applications no matter the diversity of their interaction requirements.

As noted above, we have evaluated nine selection and manipulation techniques already in the testbed environment. However, the nature of the testbed invites reuse, and we anticipate evaluating many other types of techniques in the same experimental setting. Based on our evaluations 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. When these experiments are complete, we will have a large and interesting body of performance results that will benefit the design of interactively complex VE applications.

Finally, we are interested in the process of creating formal frameworks and testbed experiments themselves. We would like to use our experience to form a set of guidelines for the development of good taxonomies, lists of metrics, and lists of outside factors. It would also be useful to have guidelines on the implementation of an effective testbed experiment. These items will be addressed in future research.

Acknowledgments

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

References

1. F. Brooks *et al.* (1992) Final Technical Report: Walkthrough Project. Report to National Science Foundation.
2. L. Hodges, B. Rothbaum, R. Kooper, D. Opdyke, T. Meyer, M. North, J. de Graff & J. Williford (1995) Virtual environments for treating the fear of heights. *IEEE Computer* 28, 27–34.
3. J. Nielsen (1993) Noncommand user interfaces. *Communications of the ACM* 36, 83–99.
4. J. Foley (1979) A standard computer graphics subroutine package. *Computers and Structures* 10, 141–147.
5. B. Price, R. Baecker & I. Small (1993) A principled taxonomy of software visualization. *Journal of Visual Languages and Computing*.
6. C. Plaisant, D. Carr & B. Shneiderman (1995) Image-browser taxonomy and guidelines for designers. *IEEE Software*.
7. S. Card, J. Mackinlay & G. Robertson (1990) The design space of input devices. In: *Proceedings of CHI*, pp. 117–124.
8. W. Barfield, D. Zeltzer, T. Sheridan & M. Slater (1995) Presence and performance within virtual environments. In: *Virtual Environments and Advanced Interface Design* (W. Barfield and T. Furness, eds). Oxford University Press, Oxford.
9. I. Poupyrev, S. Weghorst, M. Billinghurst & T. Ichikawa (1997) A framework and testbed for studying manipulation techniques for immersive VR. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 21–28.

10. D. Lampton, B. Knerr, S. Goldberg, J. Bliss, J. Moshell & B. Blau (1994) The virtual environment performance assessment battery (VEPAB): development and evaluation. *Presence: Teleoperators and Virtual Environments* 3, 145–157.
11. D. Bowman, D. Koller & L. Hodges (1997) Travel in immersive virtual environments: an evaluation of viewpoint motion control techniques. In: *Proceedings of Virtual Reality Annual International Symposium*, pp. 45–52.
12. M. Mine (1995) Virtual environment interaction techniques. UNC Chapel Hill Computer Science Technical Report TR95-018.
13. D. Bowman, D. Koller & L. Hodges (1998) A methodology for the evaluation of travel techniques for immersive virtual environments. *Virtual Reality: Research, Development, and Applications* 3, 120–131.
14. D. Bowman & L. Hodges (1997) An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In: *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pp. 35–38.
15. I. Poupyrev, M. Billinghurst, S. Weghorst & T. Ichikawa (1996) The Go-Go interaction technique: non-linear mapping for direct manipulation in VR. In: *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp. 79–80.
16. D. Bowman (1996) Conceptual design space—beyond walk-through to immersive design. In: *Designing Digital Space* (D. Bertol, ed.). Wiley, New York, pp. 225–236.
17. M. D. Bowman, J. Wineman, L. Hodges & D. Allison (1998) Designing animal habitats within an immersive VE. *IEEE Computer Graphics & Applications* 18, 9–13.
18. D. Allison, B. Wills, D. Bowman, J. Wineman & L. Hodges (1997) The virtual reality Gorilla exhibit. *IEEE Computer Graphics and Applications* 17, 30–38.
19. I. Angus & H. Sowizral (1995) Embedding the 2D interaction metaphor in a real 3D virtual environment. In: *Proceedings of SPIE, Stereoscopic Displays and Virtual Reality Systems*, Vol. 2409, pp. 282–293.