

# New Directions in 3D User Interfaces



Doug A. Bowman, Jian Chen, Chadwick A. Wingrave, John Lucas, Andrew Ray, Nicholas F. Polys, Qing Li, Yonca Haciahmetoglu, Ji-Sun Kim, Seonho Kim, Robert Boehringer, and Tao Ni

**Abstract**— Three-dimensional user interfaces (3D UIs) support user tasks in many non-traditional interactive systems such as virtual environments and augmented reality. Although 3D UI researchers have been successful in identifying basic user tasks and interaction metaphors, evaluating the usability of 3D interaction techniques, and improving the usability of many applications, 3D UI research now stands at a crossroads. Very few fundamentally new techniques and metaphors for 3D interaction have been discovered in recent years, yet the usability of 3D UIs in many real-world applications is still not at a desirable level. What directions should 3D UI researchers next explore to improve this situation? In this paper, we make some observations about the history of 3D UIs and the current state-of-the-art. Using this evidence, in addition to our own experience, we argue that 3D UI researchers should approach this problem using some new research approaches, which cluster around the concepts of *specificity, flavors, implementation, and emerging technologies*. We illustrate and discuss some of these new directions using case studies of research projects undertaken in our group. These explorations indicate the promise of these directions for further increasing our understanding of 3D interaction and 3D UI design, and for ensuring the usability of 3D UIs in future applications.

**Index Terms**—3D user interfaces, 3D interaction, user interface design, usability

## I. INTRODUCTION

With the rise of virtual environments, augmented reality, large-screen display systems, and three-dimensional (3D) applications of all sorts on the desktop, a new trend in human-computer interaction (HCI) research began to emerge. Although principles gleaned from years of experience in designing user interfaces (UIs) for desktop computers still applied, they weren't sufficient to address the unique needs of these systems where interaction took place in a 3D spatial context, with multiple degrees-of-freedom. Researchers and application developers gradually came to realize that user interfaces in these 3D arenas had some fundamental differences with traditional desktop UIs, and that new research was needed to examine interaction techniques and UI metaphors for 3D applications. This area of research gradually came to be known

as *3D User Interfaces (3D UIs)*, or *3D Interaction*.

We define **3D interaction** as “Human-computer interaction in which the user's tasks are performed directly in a 3D spatial context. Interactive systems that display 3D graphics do not necessarily involve 3D interaction; for example, if a user tours a model of a building on her desktop computer by choosing viewpoints from a traditional menu, no 3D interaction has taken place. On the other hand, 3D interaction does not necessarily mean that 3D input devices are used; for example, in the same application, if the user clicks on a target object to navigate to that object, then the 2D mouse input has been directly translated into a 3D location, and thus 3D interaction has occurred” [1]. With these concepts in mind, it is simple to define a **3D user interface** as “a UI that involves 3D interaction” [1].

Although these definitions may not allow us to precisely classify every application or interaction technique, we can further clarify them by considering the various technological contexts in which one might find 3D interaction. These include:

- **Desktop computing:** For example, users of modeling software can directly specify the 3D orientation and position of an object using a mouse in conjunction with 3D manipulation techniques.
- **Virtual environments:** For example, a user can “fly” through a virtual world through 3D pointing in the desired direction of motion.
- **Augmented reality:** For example, a physical card can represent a virtual object, allowing the object to be selected, moved, and placed in the physical world.
- **Large-screen displays:** For example, a user can zoom into a particular area of a map simply by looking at that area.
- **Ubiquitous/pervasive computing:** For example, a user can copy information from a public display to her PDA by making a gesture indicating the display and the copy action.

Research in 3D UIs has addressed a wide variety of issues. These include the design of novel 3D input or display devices [2], the empirical evaluation of user task performance with various input devices or interaction techniques [3], the development of design and/or evaluation approaches specific to 3D UIs [4], and the study of various aids to user performance such as physical props [5] or navigation aids [6], just to name a few.

The most common research topic, however, and the one we principally address in this paper, has been the design of 3D

Manuscript Received on August 25, 2006. This material is partially based upon work supported by the National Science Foundation under Grant No. NSF-IIS-0237412.

Doug A. Bowman is with the Center for Human-Computer Interaction, Virginia Tech, Blacksburg, VA 24061 USA (phone: 540-231-2058; fax: 540-231-6075; e-mail: bowman@vt.edu).

interaction techniques for the so-called “universal 3D tasks” of travel, selection, manipulation, and system control [7]. These techniques are the fundamental building blocks of 3D UIs. Their analogues in traditional UIs include techniques such as the scrollbar (navigation), point-and-click (selection), drag-and-drop (manipulation), and the pull-down menu (system control).

As we will argue in more detail later, the mid-1990s saw a boom in the discovery of such fundamental 3D interaction metaphors. This period witnessed the development of the pointing technique for 3D travel [8], the occlusion technique for 3D selection [9], the Go-Go technique for 3D manipulation [10], and the ring menu technique for 3D system control [11], among many others. But in recent years, the publication of fundamentally new 3D interaction techniques has slowed tremendously. Based on this evidence, we suggest that most of the design space for these techniques has been covered, at least at a coarse level.

From our perspective, the problem currently facing 3D UI researchers is that despite this broad coverage and extensive knowledge of 3D interaction techniques, the usability of 3D UIs in real-world applications is still surprisingly low in many cases. Thus, we argue that the community needs to consider some new directions in 3D UI design.

In this paper, we discuss some of these new directions and propose research topics whose results, we believe, will further improve the usability of 3D UIs, and further their application in the real world. These proposals come both from our own experience and from our analysis of the 3D UI literature. In particular, we argue for the potential of new research directions based on the concepts of *specificity*, *flavors*, *implementation*, and *emerging technologies*. We also illustrate several of these new directions with case studies that summarize research projects in our group.

Our goal is neither to argue that these research directions are the only useful approaches the 3D UI community should follow, nor to prove that these research directions will ultimately lead to truly usable 3D UIs. Rather, we aim to start a discussion in the community and provide researchers with some “food for thought” as they consider their future research agendas. Thus, we use informal and subjective arguments to make our case, and we do not exhaustively survey the field from every possible perspective. We welcome feedback and discussion on these issues.

## II. SELECTIVE HISTORY OF 3D UIS FROM A VR PERSPECTIVE

As a research area, 3D interaction has roots in a number of older research areas, including computer graphics, HCI, psychology, and human factors. Because 3D UI researchers rarely have a background in more than one of these areas, there are several more-or-less distinct threads that one can identify when tracing the history of 3D UIs.

For example, some human factors researchers began to work on 3D interaction when considering how to engineer training

systems for pilots, military personnel, machine operators, etc. This led to many research projects attempting to quantify human perception and performance when interacting with a virtual 3D space [12, 13].

We would argue that much of the early work on 3D UIs was *technology-driven*. That is, new interactive technologies were invented and refined, and these technologies naturally lent themselves to 3D interaction. In other words, 3D UIs arose as a means of fulfilling the requirements of other technology research areas. This assertion is difficult to prove, but we offer some subjective observations on the history of 3D UIs in virtual reality as an example.

Virtual reality (VR, also called virtual environments, VEs) is an important technology that invites 3D interaction because of its inherently spatial nature. Historically, the technologies and technological integration for VEs were realized in the late 1960s by Ivan Sutherland [14].

It is interesting to note that Sutherland’s work came from a background in computer graphics research, not HCI, although he clearly invented a new form of human-computer interface, and envisioned his work as transforming people’s visualization of and interaction with digital data [15]. Following Sutherland’s lead, work on VEs for the next 25-30 years was carried out largely by researchers with computer graphics and engineering backgrounds (although there are some notable exceptions). The focus, quite rightly, was on “making VEs work,” leading to research on real-time 3D rendering algorithms, display technology, and tracking technology. By 1994, according to Fred Brooks, VR “almost worked” in this technological sense [16].

As VR technologies improved, many attempts were made to develop real-world applications of the technologies. Although Brooks was not able to find any examples of VR technology in production use in 1994 (he excluded simulator and entertainment systems), by 1999 he identified five categories of VR applications that were being used routinely for the results they produced: architectural design and spatial layout, vehicle design, training, psychiatric treatment, and probe microscopy [17].

Initially, many of the interfaces to VR applications were designed to be “natural” – to view a virtual room the user walked around it, to fly a virtual airplane the user used real airplane controls, and to examine a virtual molecule the user picked it up directly with her hand. There are two major problems with a natural 3D UI, however.

First, because of the limitations of the technological intermediaries, natural 3D interaction cannot be completely reproduced. For example, flying the virtual airplane doesn’t feel quite right because of the mismatch between the visual and vestibular stimuli.

Second, although a naturalistic interface may be appropriate or even required for some applications, it can be extremely inefficient, frustrating, or even impractical for others. For example, in an interior design application it’s not important that the user physically picks up and moves virtual pieces of furniture with realistic force feedback. Instead, the user is only

concerned with getting the virtual furniture into the desired positions. In the same application, there are some tasks that don't lend themselves at all to natural interaction. How should the user try out a new type of desk in a virtual office design? Does he have to travel to the virtual furniture store, pick out a virtual desk, and drive it back on a virtual truck? Clearly, some form of menu system makes more sense. Such situations call for the use of a "magic" 3D UI, where the user's natural abilities and aptitudes – perceptual, motor, or cognitive – are augmented by the system. For this type of interface, unlike the more natural 3D UIs, the design space seems nearly unlimited.

As researchers and developers attempted to create applications with complex, magic interaction, a crisis emerged. It was obvious that magic 3D interaction could greatly benefit many types of VR applications, but how should such interaction work? How can a VE user travel thousands of virtual miles quickly and easily and still end up at the desired location? What is the best way to select an item from a menu in a 3D context? 3D UI design presented many new challenges and difficulties [18], even when existing HCI knowledge was considered, and most of the application prototypes using magic 3D interaction exhibited significant usability problems. This led Brooks to list 3D interaction as one of the most crucial research areas that would affect the speed of adoption and success of VR [17].

Starting in approximately the early- to mid-1990s, then, we observed that research in 3D UIs (and 3D interaction techniques in particular) boomed. Research focused on tasks that were present in most VR applications. These are the "universal tasks" mentioned earlier: travel, selection, manipulation, and system control [7].

The "3D interaction boom" continued through the 1990s, with a large number of novel interaction techniques published in conferences and journals in the HCI, graphics, human factors, VE, and augmented reality (AR) communities. In addition, many empirical studies of the usability of 3D interaction techniques were published during this period, and researchers developed classifications of 3D interaction tasks and techniques as well. By the year 2000, however, we feel that the boom began to slow significantly. Although there were still many researchers studying 3D UIs, the number of novel 3D interaction techniques introduced to the community decreased.

To quantify this argument, we analyzed the publication dates of the interaction techniques presented in the book *3D User Interfaces: Theory and Practice* [1]. Although not all published techniques are discussed, the best-known, most widely-cited, and most widely-used techniques are represented. Although there is certainly some subjectivity involved, we feel that the book's coverage is currently the best representation of the state-of-the-art in 3D interaction techniques. We considered citations only from chapters 5 (selection and manipulation), 6 (travel), and 8 (system control). Where a technique had more than one citation, we used the earliest one.

The results of our analysis are presented in Fig.1. As the figure shows, the distribution of publication dates resembles a bell-shaped curve. The majority of the novel interaction

techniques cited fall between 1994-1999 (37 out of 53 or 69.8%) with over 20% published in 1995 alone. In addition, five out of the six most cited years occurred in that span. In the five years immediately prior to the book's publication (2000-2004), only six techniques were deemed important or novel enough for inclusion.

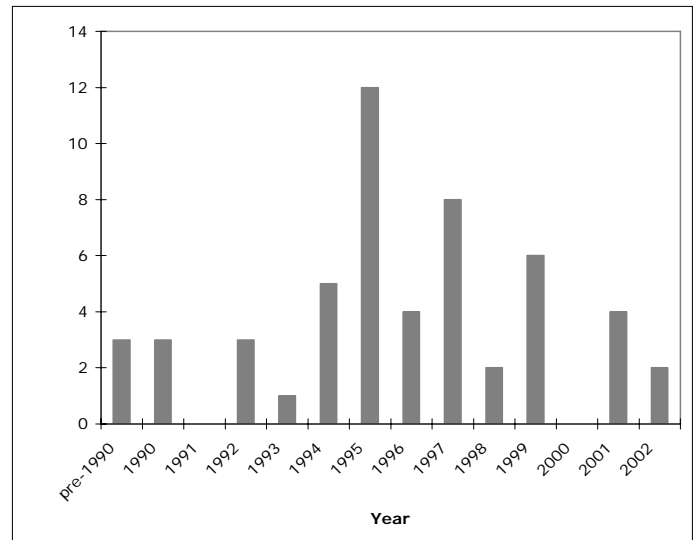


Fig. 1. Number of novel 3D interaction technique publications cited by year in *3D User Interfaces* [1]

Obviously, this analysis does not provide conclusive evidence for our argument, nor does not tell us what will happen in the future. There may be fundamental new 3D interaction techniques still invented, but the trend does seem to indicate that many of the most basic techniques for the universal 3D interaction tasks have been discovered and implemented.

### III. THE SITUATION TODAY

So where does that leave us? If we agree that the fundamental techniques for the universal 3D interaction tasks have been discovered already, what should happen next in the 3D UI research area? There does not seem to be a lack of interest in this topic, despite the downward trend in the discovery of new interaction techniques. The 3DUI mailing list ([www.3dui.org](http://www.3dui.org)), a worldwide community of researchers in this area, currently has over 350 members from at least 28 countries. Attendance at workshops, tutorials, and presentations on 3D UI-related topics is consistently high in our experience. The first international symposium on 3D UIs, sponsored by the IEEE, was held in 2006. But how should all of these researchers focus their efforts? What still needs to be done in 3D UI research? To answer these questions, we suggest that it is instructive to consider the current state of 3D applications.

If all of the important 3D interaction techniques have been discovered, one might expect to find a corresponding boom in usable, real-world applications involving 3D interaction. Unfortunately, this does not seem to be the case. Although

there has not been another systematic study on production applications of VEs since Brooks' report in 1999, anecdotal evidence suggests that there are only limited, incremental successes, many of which are similar to the categories of applications discovered by Brooks. Furthermore, most of the production applications of VR have only simple or natural 3D interaction, or no 3D interaction at all (VE applications used for visualization and exploration of geological models in the oil & gas industry are a notable exception). Our experience with demonstrations of highly interactive 3D application prototypes is that there are still significant usability problems in most cases. Overall, then, we have not seen any evidence that there has been a significant increase in usable real-world applications involving complex 3D interaction.

This lack of an application boom could be interpreted in several ways. First, it could be that there are fundamental 3D interaction techniques that have still not been invented, and that the use of these techniques would produce usable 3D UIs. Although this is possible, we have argued in the previous section that it is not likely. Second, the results of 3D UI research may not have reached the application developers, or the developers may have ignored the research results. In other words, although the knowledge exists to develop usable 3D UIs, this knowledge is not being used to its full potential. We believe that this interpretation has some merit, and that 3D UI researchers need to continue to push their results out of the laboratory and into the real world. Third, organizations may not be convinced that 3D interaction and associated technologies like VR have any commercial benefits. This explanation also likely has some truth to it, but it does not explain the usability problems of even the prototype applications.

There is a fourth possible interpretation: it may be that most or all of the basic 3D interaction techniques have been discovered, but that this knowledge alone is not sufficient to ensure the usability of 3D UIs. In other words, even if everyone developing applications with 3D interaction had access to all the most important research results, the applications would still have usability problems. What follows from this argument is that we need to discover what other types of knowledge are needed about 3D UIs – we need an agenda outlining the next steps in 3D UI research. The remainder of the paper is devoted to our take on this topic.

#### IV. PROPOSED NEW DIRECTIONS

In this section we propose a research agenda to move 3D UI research beyond the design and evaluation of 3D interaction techniques for the universal 3D tasks. To address the question of which directions 3D UI research should take next, we relied on our own experience in developing and evaluating 3D interaction techniques and complete 3D UIs [19, 20, 21], and on our experience with the usability of prototype 3D applications with high levels of interactivity [22, 23, 24]. We also examined the most recent work by researchers in the area, as exemplified by the proceedings of two recent workshops on

3D UIs [25, 26] and the first IEEE Symposium on 3D UIs [27], for clues about potential new directions. Our proposed agenda centers on four research directions:

- Increasing *specificity* in 3D UI design
- Adding, modifying, or tweaking 3D interaction techniques to produce *flavors*
- Addressing the *implementation* issues for 3D UIs
- Applying 3D UIs to *emerging technologies*

##### A. *Specificity*

Although we have argued that most or all of the fundamental 3D interaction techniques for the universal tasks have been discovered, we do not claim that researchers should stop working on interaction techniques altogether. Rather, we believe that existing 3D interaction techniques do not provide sufficient usability in many real world applications because of the *overgenerality* of these techniques.

Typically, generality is considered to be a desirable characteristic. If something is more general, it can be applied in more ways. For example, because the ray-casting technique addresses the general task of 3D selection, it can be used in any situation requiring 3D selection. The downside of this generality, however, is that the application of a technique is always to a *specific* situation, and a general technique has not been designed for the specific requirements of that situation.

The large majority of the 3D interaction techniques we surveyed in section II exhibit generality in at least four different ways.

- **Application- and domain-generality:** The technique was not designed with any particular application or application domain in mind, but rather was designed to work with any application.
- **Task-generality:** The technique was designed to work in any task situation, rather than being designed to target a specific type of task. For example, the design of the ray-casting technique does not take into account the size of the objects to be selected and becomes very difficult to use with very small objects [28].
- **Device-generality:** The technique was designed without consideration for the particular input and display devices that would be used with the technique. Often techniques are implemented and evaluated using particular devices, but the characteristics of those devices are not considered as part of the design process. For example, the HOMER technique [19] is assumed to work with any six-degree-of-freedom input device and any display device, but all of the evaluations of this technique (of which we are aware) have used a wand-like input device and a head-mounted display (HMD).
- **User-generality:** The technique was not designed with any particular group of users or user characteristics in mind, but rather was designed to work for a "typical" user.

To improve the usability of 3D interaction techniques in real-world applications, we propose that researchers should consider *specificity* in their designs. The types of specificity we

have identified are parallel to the types of generality described above:

- **Application-specificity:** The most obvious way to improve usability in any particular application is to design the interaction techniques and user interface specifically for that application. This would mean basically starting from scratch with each application and using a complete usability engineering process [29] to produce a usable 3D UI. The disadvantage of this, of course, is that it's *too specific*. There is no guarantee that any interaction techniques or UIs produced by this method can be reused in any other application.
- **Domain-specificity:** To balance the advantages and disadvantages of general and specific 3D UI design, we propose the concept of *domain-specific* 3D interaction techniques and UIs [30]. In this approach, designers acquire knowledge about a particular application domain, and then use that knowledge in their UI designs. The resulting interaction techniques and UI components should be reusable in multiple applications in the same domain. For a case study on the design of domain-specific 3D UIs, see section V.A.
- **Task-specificity:** Techniques can also be made more usable by designing for specific task requirements rather than for a general task category. For example, many general 3D travel techniques exist, but very few researchers have designed travel techniques for specific types of travel tasks, such as exploration, targeted search, or maneuvering. In addition, there are specific 3D interaction tasks that should be considered separately from the universal tasks. For example, 3D surface creation and modification, though similar to object manipulation, is a task that would benefit from task-specific techniques. We present a case study of task-specific technique design in section V.B.
- **Device-specificity:** We can also increase the level of specificity relative to particular input and/or display devices. A technique or UI designed for a pen-shaped 6-DOF input device with a single button may exhibit serious usability problems when used with a multi-button 6-DOF SpaceBall™, for example. There are also numerous opportunities to exploit display device-specificity. In particular, many of the fundamental 3D interaction techniques were designed and evaluated for either desktop displays or HMDs, but we have little or no data on the usability of those techniques with surround-screen or large-screen displays. A case study on display-specific 3D UI design can be found in section V.C.
- **User-specificity:** Finally, techniques may be made more specific by considering specific groups of users or particular user characteristics in the design. For example, VE experts may use different strategies, and would therefore benefit from an expert-specific interaction technique [31]. User-specific techniques could be created on the basis of age, expertise, gender, spatial ability, and

the like.

There is a great deal of potential research in the area of specificity. In particular, it will be important for us to understand when the benefit of designing more specific 3D UIs outweighs the extra cost.

### B. *Flavors of 3D interaction techniques*

Existing 3D interaction techniques are, for the most part, simple and straightforward. They are designed to do one thing, and to do it well. In some cases, however, usability can be improved by adding features or complexity to one of these basic techniques. The concept of *flavors* of 3D interaction techniques refers to such variations on fundamental techniques. Often such flavors can be designed by considering lists or classifications of usability problems discovered in fundamental techniques.

For example, consider the basic ray-casting technique [8]. Its design is extremely simple, and users find the metaphor clear and easy to understand. However, many researchers have noted shortcomings of this technique for the task of 3D object selection [28]. Adding features such as snapping the selection ray to the nearest object [31] or bending the ray around occluding objects [32] could address some of these usability problems.

It is not clear which flavors will be useful, and what the tradeoff will be between the level of simplicity of a basic technique and the power of a “flavored” technique. In addition, the added complexity of flavors leads to difficult implementation issues (see the next section). All of these are topics for future research. For a case study on flavors, see section V.D.

### C. *Implementation of 3D UIs*

To a large degree, the 3D UI community has ignored the problem of implementation and development tools, although this problem has been recognized [18, 33]. Most techniques are implemented using existing VE, AR, or 3D graphics toolkits, and there is no standard library of 3D UI components available. For basic techniques, this may not be too great a burden. However, as we move toward more complex 3D UIs (including those resulting from the specificity or flavors approaches), turning a good design into a good implementation becomes more and more difficult.

There are many factors that make 3D UI implementation problematic, including the following:

- 3D UIs must handle a greater amount and variety of input data.
- There are no standard input or display devices for 3D UIs.
- Some input to 3D UIs must be processed or recognized before it is useful.
- 3D UIs often require multimodal input and produce multimodal output.
- Real-time responses are usually required in 3D UIs.

- 3D interactions may be continuous, parallel, or overlapping.

3D UI developers must manage all of this complexity while using tools and methods that were originally intended for simple 2D desktop UI programming, using 3<sup>rd</sup>-generation languages that lack the flexibility of more modern scripting languages. The result is code that is very hard to design, read, maintain, and debug; code that contains a large number of callback functions with complicated interrelationships; and code that makes inordinate use of global variables as developers try to manage the state of the application.

Moreover, as HCI researchers, we know that the ability to perform rapid design iterations on low- or medium-fidelity prototypes is an invaluable part of the UI development process. But no tools for the rapid prototyping of 3D UIs currently exist.

To enable further research in 3D interaction, therefore, researchers must address these implementation issues. One important step, though not a complete solution, would be a standard library of interaction techniques or technique components. Such a library would contain generic, reusable implementations of fundamental interaction techniques for the universal tasks. To be truly useful, such a library would need to interoperate with other toolkits used for developing 3D applications. A second major research question is how to enable the rapid prototyping of 3D UIs – both those using existing techniques and those with novel interaction methods. Finally, we need to address the development tools themselves – current 3D graphics development tools based on procedural programming and callbacks may not be the most appropriate choice for 3D UIs. A case study describing some initial steps in this direction can be found in section V.E.

#### D. 3D UIs in emerging technologies

Finally, we propose that technological changes should stimulate additional 3D UI research. Up to this point, the majority of 3D UI research has targeted desktops, VEs, or AR. But there are a number of emerging technologies or technological concepts, such as large-display technology, wide-area tracking technology, and pervasive computing technology, which will also provide an opportunity for further 3D UI research.

Perhaps the best current example is the trend towards visual displays that are larger in size and that display a much greater number of pixels than traditional displays. Specific systems range from multi-monitor desktop displays to large wall-sized displays to huge public displays. All of these systems could make use of some form of 3D interaction, even if the display is showing 2D information. For example, in the multi-monitor desktop case, losing the mouse pointer is a common problem [34]. Using a 3D input device allowing the user to point directly at the screen in some way might alleviate this problem [35]. In the very large display case, users cannot physically reach the entire display, so interaction techniques based on physical walking, gaze direction, gestures, and the like may be appropriate [36].

Such emerging technologies provide an important test of the quality of existing 3D UI research. Fundamental principles and techniques developed by 3D UI researchers working with VEs (for example) should be applicable to some degree for large-screen displays (for example). In particular, researchers should address questions like: “What are the universal 3D interaction tasks for the emerging technology, and do they match the universal tasks already identified?” “Which existing 3D interaction techniques can be repurposed in applications of the emerging technology?” and “What changes are required to migrate existing 3D UI techniques and principles to the emerging technological area?” This type of research should prove to be fruitful and challenging in the next few years. We offer a case study of this direction in section V.F

## V. CASE STUDIES OF NEW DIRECTIONS

We now turn to examples of the new directions discussed in section IV. All of these projects were carried out in the 3DI Group at Virginia Tech, and several come from a graduate class on 3D interaction. In the sections that follow, we summarize each of the projects. Many details are left out; we only provide enough information to give a sense of the problem, the proposed solution, the experimental results, and the overall approach taken by each project. Readers interested in more information can look into the papers referenced at the end of each section.

### A. Domain-specificity: Cloning techniques for the AEC domain

As a testbed for the concept of *domain-specific 3D interaction*, we considered the architecture / engineering / construction (AEC) domain. In the past, we have designed several 3D UIs for particular applications in this domain [22, 24, 37], and we have attempted to use generic 3D interaction techniques within those UIs. Although we achieved moderate success in the usability of the applications, some significant usability problems could not be solved using generic techniques. We hypothesized that by learning more about the AEC domain and applying this knowledge to our 3D UI designs, we could both improve the usability of particular applications and, at the same time, design techniques that could be reused elsewhere in the domain.

This project has produced some significant results, but not in the way we originally expected. We began by gathering as much information about the AEC domain as we could, and trying to extract from this information particular characteristics of the domain that might be relevant to 3D interaction. For example, it's clear that because the AEC domain deals with physical objects such as columns, walls, and furniture, that domain-specific object manipulation techniques need to take the physical properties of these objects into account (e.g. a chair must rest on a floor and should only rotate around its vertical axis). We quickly developed a short list of such characteristics, but most of them could be reduced to simple

*constraints*, which are already widely used in 3D UIs [38, 39]. To have real impact on the field, we needed another method for designing domain-specific 3D interaction techniques.

Our most significant results came not from considering domain characteristics in the abstract, but rather from considering the requirements of a particular application in the domain. The Virtual-SAP application [24] is a tool for structural engineering students and practitioners. It allows them to design and build the structural elements of a building, and then simulate the effects of environmental conditions, such as earthquakes, on the structure. The original 3D UI for this application used generic interaction techniques such as pointing for travel, the Go-Go technique for selection and manipulation, a 3D snap-to grid to aid precise manipulation, and pen-and-tablet-based menus. These techniques were all relatively usable in our studies of engineering and architecture students. What we discovered, however, was that the *tasks* supported by the application were not sufficient for real-world work. In particular, users had trouble creating structures with sufficient complexity and size, because the UI required them to create each element of the structure via the menu, and then place it in the environment using the manipulation technique. For a structure with hundreds of elements, this became impractical. This problem was a significant one, because students and practitioners really only benefit from the application by analyzing complex cases where they could not predict the behavior of the structure.

We observed that most of the large structures users wanted to build contained a great deal of repetition (e.g. one floor of a building is identical to the one below it), and so we addressed this problem by developing interaction techniques for the task of *3D cloning*. These techniques allow the user to quickly create multiple copies of existing components of a structure. We went through many design iterations and now have several usable and efficient techniques [40; see Fig.2], but the key was recognizing that the AEC domain required a 3D interaction task we had not originally considered.

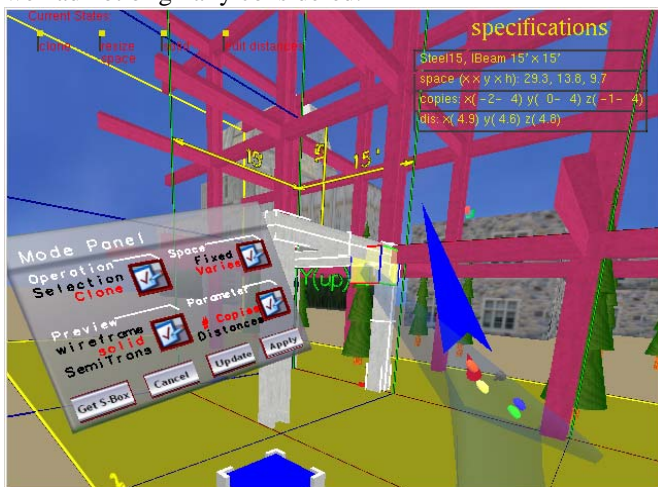


Fig. 2. One of the 3D cloning techniques designed through domain-specific interaction research. See Color Plate 1

In addition, our work on cloning techniques required us to

consider another 3D interaction task – multiple object selection. In order to clone an existing component of a structure, the user must select all the elements (beams, walls, etc.) that make up that component. Of course, selecting the elements in a serial fashion with generic 3D selection techniques would address this requirement, but just as creating multiple copies is more efficient than creating one element at a time, selecting many objects in parallel may provide performance gains. We have developed and evaluated both serial and parallel multiple object selection techniques [41], and we are currently using one of the parallel techniques (a 3D selection box) in our cloning interface.

In summary, we have developed several successful domain-specific interaction techniques that provide clear usability gains over their generic counterparts [30]. These techniques can be reused in multiple AEC applications, and may also be useful in other domains. But the process of developing these techniques has differed from our expectations in two important ways. First, real applications in the domain were a better source of UI requirements than general knowledge about the domain. Second, designing new techniques for domain-specific interaction tasks has been more successful than designing domain-specific interaction techniques for universal interaction tasks, at least so far. For more information, see [30, 40].

### B. Task-specificity: resizing techniques

A second project focused on the task of resizing, or scaling, virtual objects. In fact, this project was motivated by the 3D multiple object selection project described in the previous section. In the 3D selection box technique, the user encloses the objects she wants to select with a semi-transparent box. In order to use this technique for any set of contiguous objects, the user must be able to both position and resize the box. In a way, then, our consideration of this interaction task also arose from our examination of the AEC domain.

The scaling of virtual objects, of course, is not a new 3D interaction task. Prior research on this task has resulted in a number of widget-based techniques [42, 43]. In the widget-based approach, small handles are attached to the selected object, and the user scales the object by manipulating one or more of the handles. The key word in the previous sentence is *manipulating* – often 3D UI designers have viewed the resizing task as simply a special case of manipulation, and have therefore simply reused existing manipulation techniques to position the widgets.

Our approach, on the other hand, was to consider the unique properties of the resizing task as an opportunity to do some focused design; in other words, we took a *task-specific* approach. We are not the first to take this approach for object resizing. For example, there are several published techniques using two-handed interaction to specify object scale [44, 45]. Still, there are many other novel designs that have not yet been tried.

We designed object-resizing techniques for immersive VEs



using an HMD. Our work resulted in two novel techniques, both of which require only one tracked hand for interaction. The Pointer Orientation-based Resize Technique (PORT) uses the orientation of the hand-held pointer relative to the object to determine the axis of scaling, and the amount of scaling is specified using a small joystick (Fig.3). With the Gaze-Hand Resize technique (Fig.4) the user specifies the axis of scaling using the orientation of the pointer as in the PORT technique. The user's gaze direction is then used to adjust the position to which the active face of the box will be moved. This technique takes advantage of the stability of the head to perform accurate resizing of objects.

We hypothesized that both of our novel techniques would be more efficient and more accurate than a widget-based technique for resizing an object along a single axis. However, our techniques also had a disadvantage: they only allowed scaling along one axis at a time, while our implementation of the widget-based technique allowed the user to scale along one, two, or all three axes simultaneously by choosing an appropriate widget.

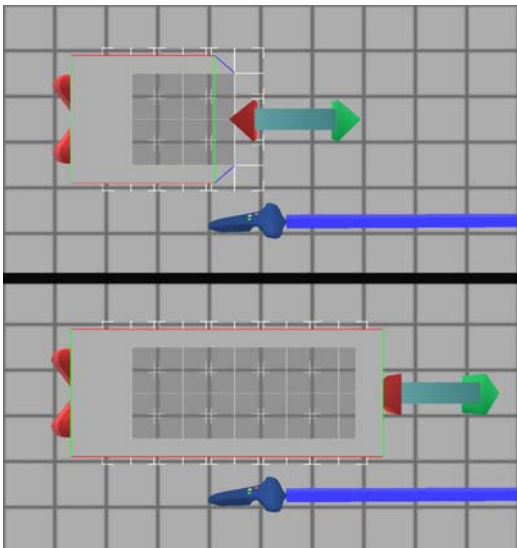


Fig. 3. Resizing a box with the PORT technique

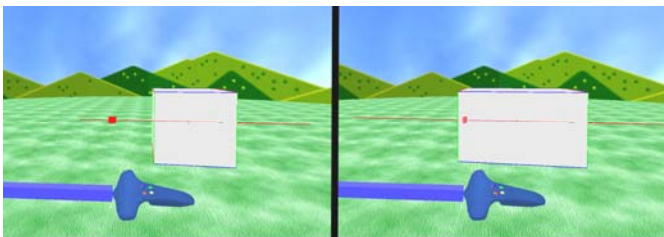


Fig. 4. Resizing a box with the Gaze-Hand Resize technique

We evaluated our two techniques and the widget-based technique in a formal experiment. Both PORT and Gaze-Hand were significantly faster than 3D widgets. Initially, subjects preferred the 3D widgets because of their similarity to resizing techniques in desktop applications, but by the end of the experiment, the strongest preference was for the Gaze-Hand

technique, while 3D widgets was the least preferred.

This project demonstrates the potential of the task-specific approach to 3D UI design. Designing for the specific task situation resulted in higher levels of objective and subjective usability as opposed to simply reusing more generic interaction techniques. Complete details and results of this research can be found in [46].

### C. Display-specificity: IRVE information layout techniques for different displays

One of the current research thrusts in our group focuses on *information-rich virtual environments* (IRVEs), which combine realistic perceptual/spatial information with related abstract information [47]. One of the important research questions related to IRVEs is how best to layout the abstract information relative to the spatial environment. For example, information may be in *object space* (within the 3D environment) or *viewport space* (overlaid on the user's view of the environment), among other choices. Our previous research has compared these two layout options for desktop displays, but we were interested in how these layout techniques scaled to larger displays.

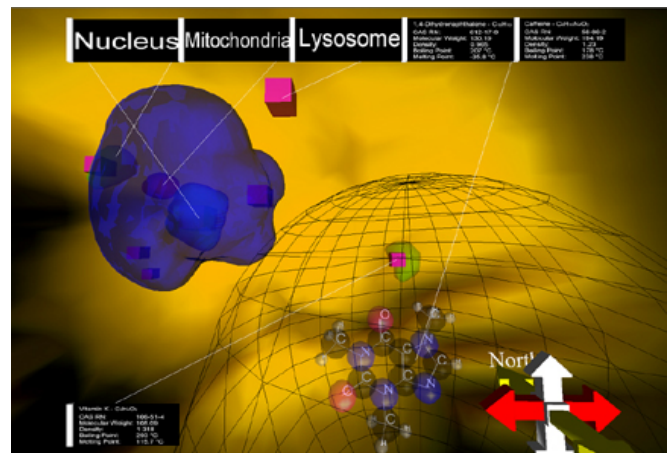
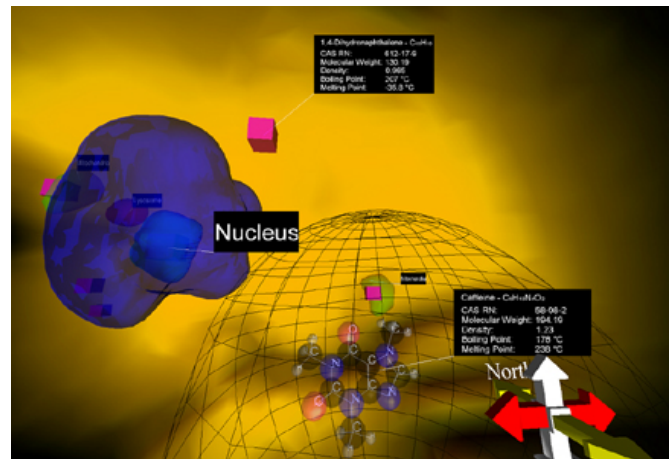


Fig. 5. Layout techniques for IRVEs: object space (top) and viewport space (bottom) layouts. See Color Plate 2 & 3

In this study, we compared a typical desktop monitor with a



large, high pixel-count display made up of nine tiled LCD panels in a 3x3 configuration. We displayed an IRVE representing a biological cell, with abstract information labeling various parts of the cell and their properties. The object space layout technique (Fig.5, top) placed this abstract information on text panels that were part of the 3D environment and that were attached to the appropriate 3D objects by lines. The viewport space technique was a heads-up display (HUD) in which the abstract information was placed around the border of the display, in front of the environment (Fig.5, bottom). The HUD information was also connected to the 3D objects with lines. The labels in the HUD condition have a fixed size measured in pixels. Thus, in the large display condition many more labels can be viewed simultaneously without any loss of legibility; this was thought to be a significant advantage of the large display.

Overall, the HUD technique was more effective and preferred when subjects performed search and comparison tasks in the IRVE. However, a surprising effect was that some of the HUD's advantages were lost when the large display was used, while performance with the object space layout increased with the large display. Our interpretation of this result is that because we chose to place labels on the border of the display in the HUD technique, users of the large display were forced into greater head and eye movements, and labels ended up farther from their corresponding 3D objects.

This example shows that well-designed and usable techniques for one display may not migrate well to different display devices. In this case, the HUD concept clearly needed to be redesigned for the large display. For example, we might decide to place the labels on the image plane near to their corresponding 3D objects when using a large display, similar to the technique used in [48]. Further research along these lines is needed to determine the effects of display type on a variety of 3D interaction techniques and UI designs. For more detail on this research, see [49].

#### D. Flavors: The SSWIM technique

The *flavors* approach adds new features, additional complexity, or subtle tweaks to existing fundamental techniques for the purpose of improving usability. An example of this approach is the SSWIM technique – an enhancement of the well-known World-in-Miniature, or WIM [50, 51].

The WIM technique gives the user a hand-held miniature version of the virtual world. The user can select and manipulate objects in the world by directly touching and moving their miniature representations in the WIM. The WIM can also be used for travel – the user grabs and moves a miniature representation of herself in order to travel to a new location in the full-scale environment.

One of the major limitations of the WIM is its lack of scalability. In a very large environment, either the WIM will be too big to handle, or the miniature versions of the object will be too small to see and manipulate easily. Thus the simple concept of this project was to allow the user to “pan and zoom” the

WIM's representation of the virtual world – we call this the “Scaled Scrolling World-in-Miniature” or SSWIM. LaViola's Step WIM [52] also added the ability to zoom, but this design deviated significantly from the original WIM – the miniature was fixed to the floor and users interacted with their feet. Our goal was to remain true to the original WIM concept while adding more advanced features.

The SSWIM (Fig. 6) was designed for an HMD-based system where both of the user's hands are tracked. The SSWIM itself is held in the non-dominant hand and is automatically rotated to remain aligned with the full-scale environment. A wand is held in the dominant hand and is used to reposition the user representation within the SSWIM. Scrolling is accomplished by dragging the user representation towards the edge of the SSWIM, and a large arrow provides feedback as to the direction of scrolling. The most difficult design decision was how to allow users to zoom the miniature world. Initially we tried a glove-based approach where the user's hand posture (somewhere on the continuum between a clenched fist and a flat hand) determined the scale, but this was too difficult to use and control. We next tried scaling up or down with two additional buttons on the wand, but this prohibited the user from scaling the miniature and positioning the user representation at the same time. Eventually we hit upon the idea of using the scroll wheel of a wireless mouse held in the non-dominant hand for scaling.

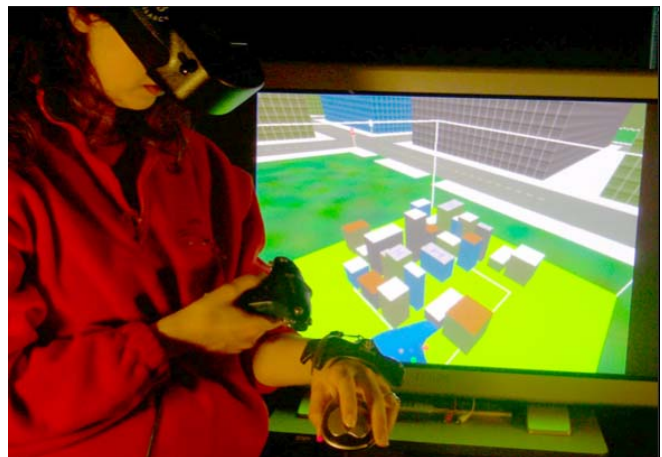


Fig. 6. The SSWIM technique. See Color Plate 4

Obviously, the additional features of the SSWIM make it possible to travel precisely even in environments with a wide range of scales. For example, in the city environment we used, it would be possible to travel from one side of the city to the other, and to enter a specific room within one of the high-rise buildings. Our question, however, was whether this additional complexity would cause a significant drop in usability for the travel tasks already supported by the WIM. We compared the two techniques in an experiment designed to test typical WIM travel tasks, and found no significant difference in efficiency or user preference. In fact, SSWIM users were significantly more accurate than WIM users in positioning themselves in the world,

because they consistently chose to use a zoomed-in miniature world representation.

This project demonstrates that it is possible to create usable new flavors of existing fundamental 3D interaction techniques that provide significant advantages. It also shows that the additional complexity does not necessarily lead to a drop in user performance, even on simple tasks. A complete description of the SSWIM technique and the user study is in [53].

The other major lesson from this project, however, was that the implementation of such complex techniques is extremely difficult with existing tools. This has led us to consider alternative 3D UI implementation approaches (see the following section).

#### *E. Implementation framework for 3D UIs: CHASM*

As the previous examples show, our research into domain-specific, task-specific, device-specific, and flavored 3D interaction techniques allowed us to generate a wide variety of new ideas for improving the usability of 3D UIs. Clearly there is no shortage of novel contributions that can be made to 3D UI design. But a consistent subplot to our research has been our frustration in implementing the envisioned techniques and interfaces. As we noted above, it is extremely difficult with today's development tools to produce a robust, bug-free implementation of even a moderately complex 3D interaction technique. Our research group has a great deal of experience with 3D graphics, VE toolkits, and UI programming using an event-based or simulation-based model; still, we find it troublesome to translate our envisioned 3D UIs into working code.

We have evaluated 3D UI designers' natural-language descriptions of their interfaces and interaction techniques. Our analysis shows that there is a large gap between these descriptions and the required implementations using today's tools. The descriptions do seem to match more closely, however, with a state machine representation of interaction. Therefore we have developed an implementation framework based on state machines. This framework is called CHASM (Connected Hierarchical Architecture of State Machines) [54].

CHASM addresses the issues with event-based programming by using a state machine representation, but we quickly realized that simple state machines would not be sufficient. As UIs grow in complexity, there is a state space explosion that would be even more difficult to manage than the callbacks in traditional UI programming. Thus, CHASM uses a hierarchy of state machines, where each machine can be partitioned into manageable chunks, and where the designer can work at various levels of abstraction. CHASM provides a GUI allowing users to specify these state machines visually. CHASM also supports and encourages reuse, since each state machine is self-contained and modular.

We have just begun to use CHASM for actual 3D UI development, but the preliminary results are encouraging. While CHASM itself still has a steep learning curve, the

techniques it produces are more understandable, more robust, easier to modify, and easier to reuse.

CHASM is not the only possible approach for enabling more rapid and robust development of 3D UIs, but it is certainly promising. We hope that it will inspire others in the 3D UI community to investigate methods for addressing the difficult implementation issues related to 3D interaction. For more information on this project, see [54].

#### *F. Emerging technologies: 3D input for large displays*

A recent trend outside the VR community is the use of very large, high-resolution displays for information visualization, geospatial exploration, visual analytics, and the like. Most of this data is 2D in nature, but because traditional 2D interaction techniques (e.g. drag-and-drop) are often less than usable on such displays, we see an opportunity for 3D interaction in this area.

We have been exploring the use of 3D input devices and interaction techniques for large displays. A recent project in this direction investigated the use of 3D tracking technology as an input device for navigating geospatial visualizations (Fig. 7). A tracker, worn on the head or held in the hand, is used to pan the imagery shown on the display. To pan the imagery to the left, the user looks or points to the right, and vice-versa. Vertical panning is also possible.

A preliminary study showed that these techniques were promising, and that they worked especially well with the largest display sizes. The handheld tracker technique performed just as well as or better than techniques using traditional mouse-based input. It was preferred to the head-worn tracker technique, perhaps because it was more flexible and less fatiguing.

We are continuing to explore this research direction, but there is clearly a great deal of promise in the application of 3D interaction techniques to technologies other than VR or AR, even when the displayed data is not 3D.

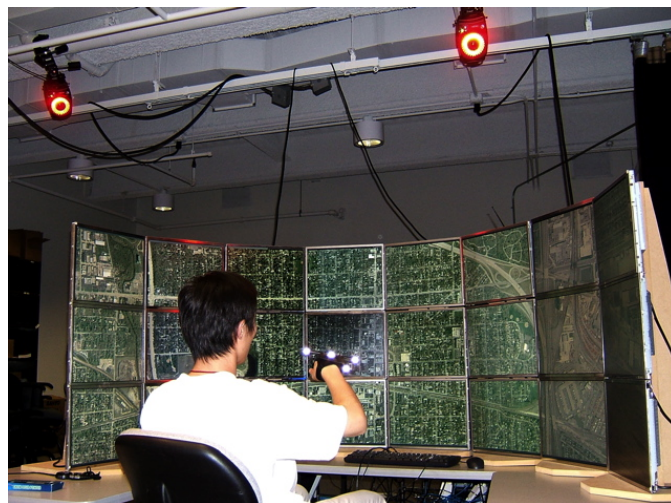


Fig. 7. 3D tracker for large-display interaction. See Color Plate 5

## VI. CONCLUSIONS

The topic of 3D UIs has led to a great deal of interesting research. However, the pace of growth in the area seems to have slowed significantly since the year 2000 or so. At the same time, we have not seen a significant increase in the quantity or quality (usability) of real-world VR/AR applications due to 3D UI research. Thus, we feel that the area is in need of some new research directions that can address this situation.

We have proposed that research topics related to *specificity* (explicit consideration of specific domains, tasks, and devices), *flavors* (adding complexity, features, or tweaks to existing techniques), *implementation* (enabling the rapid and robust development of 3D UIs), and *emerging technologies* (applying or redesigning existing 3D UIs for new technological contexts) will have a major positive impact on the field and on the usability of real-world applications of 3D UIs. Our case studies of several of these new approaches indicate the potential they have for improving 3D interaction design and development.

Of course, we have not covered all possible new directions in this paper. For example, the emerging theme of *embodiment* in HCI certainly applies to 3D UI design, where users are often bodily immersed in a real or virtual 3D space. Another important research topic centers on the *integration* of disparate 3D interaction techniques into a complete 3D user interface. And of course, these new directions are not entirely new – some of them have been used in other areas of HCI quite successfully. Based on the research presented at the most recent international workshops and symposia on 3D UIs, some 3D interaction researchers are already beginning to use these approaches. But they do represent a change from the traditional ways of doing 3D UI research. We hope that the directions outlined here can serve as a useful starting point for 3D UI researchers just entering the field or those looking for a new challenge.

Taking these approaches as a research agenda should help to increase our knowledge and understanding of 3D UI design. We look forward to exciting results based on these new directions, and eventually to usable 3D UIs in a wide range of real-world applications.

## ACKNOWLEDGEMENT

The authors gratefully acknowledge the entire 3D UI research community, whose work inspired and informed this paper. In particular, we would like to thank Bernd Fröhlich, Yoshifumi Kitamura, and Wolfgang Stürzlinger for their work on the 3D UI workshops, and Ernst Kruijff, Joe LaViola, and Ivan Poupyrev for their long-term collaboration.

## REFERENCES

- [1] D. Bowman, E. Kruijff, J. LaViola and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Boston: Addison-Wesley, 2004.
- [2] B. Froehlich and J. Plate. The Cubic Mouse: A New Device for Three-Dimensional Input, *Proceedings of ACM CHI*, 2000.
- [3] K. Hinckley, J. Tullio, R. Pausch, D. Proffitt and N. Kassell. Usability Analysis of 3D Rotation Techniques, *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp. 1-10, 1997
- [4] D. Hix and J. Gabbard. Usability Engineering of Virtual Environments, in *Handbook of Virtual Environments*, K. Stanney, Ed.: Lawrence Erlbaum, pp. 681-699, 2002
- [5] D. Schmalstieg, L. Encarnacao and Z. Szalczvari. Using Transparent Props For Interaction with The Virtual Table, *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pp. 147-154, 1999
- [6] R. Darken, H. Cevik. Map Usage in Virtual Environments: Orientation Issues, *Proceedings of IEEE Virtual Reality*, pp. 133-140, 1999
- [7] D. Bowman, E. Kruijff, J. LaViola and I. Poupyrev. An Introduction to 3D User Interface Design, *Presence: Teleoperators and Virtual Environments*, vol. 10, pp. 96-108, 2001
- [8] M. Mine. Virtual Environment Interaction Techniques. UNC Chapel Hill CS Dept., *Technical Report TR*, pp.95-018, 1995.
- [9] J. Pierce, A. Forsberg, M. Conway, S. Hong, R. Zeleznik and M. Mine. Image Plane Interaction Techniques in 3D Immersive Environments, *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pp. 39-44, 1997
- [10] I. Poupyrev, M. Billinghurst, S. Weghorst and T. Ichikawa. The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR, *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp. 79-80, 1996
- [11] J. Liang and M. Green. JDCAD: A Highly Interactive 3D Modeling System. *Computer & Graphics*, vol. 4, pp. 499-506, 1994.
- [12] W. Barfield, C. Hendrix and K. Bystrom. Visualizing the Structure of Virtual Objects Using Head Tracked Stereoscopic Displays, *Proceedings of the Virtual Reality Annual International Symposium*, pp. 114-120, 1997
- [13] S. Zhai, P. Milgram. Human Performance Evaluation of Manipulation Schemes in Virtual Environments, *Proceedings of the Virtual Reality Annual International Symposium*, pp. 155-161, 1993
- [14] I. Sutherland. A Head-mounted Three Dimensional Display, *Proceedings of the Fall Joint Computer Conference*, pp.757- 764,1968
- [15] I. Sutherland. The Ultimate Display. *Proceedings of the IFIP Congress*, pp. 505-508, 1965
- [16] F. Brooks. Is there any real virtue in virtual reality? , London, 1994.
- [17] F. Brooks. What's Real About Virtual Reality? , *IEEE Computer Graphics & Applications*, vol. 19, pp. 16-27, 1999.
- [18] K. Herndon, A. van Dam and M. Gleicher. The Challenges of 3D Interaction, *SIGCHI Bulletin*, vol. 26, pp. 36-43, 1994.
- [19] D. Bowman and L. Hodges. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments, *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pp. 35-38, 1997
- [20] D. Bowman and C. Wingrave. Design and Evaluation of Menu Systems for Immersive Virtual Environments, *Proceedings of IEEE Virtual Reality*, 2001, pp. 149-156.
- [21] D. Bowman, C. Wingrave, J. Campbell, V. Ly and C. Rhoton. Novel Uses of Pinch Gloves for Virtual Environment Interaction Techniques, *Virtual Reality*, vol. 6, pp. 122-129, 2002.
- [22] D. Bowman, J. Wineman, L. Hodges and D. Allison. Designing Animal Habitats Within an Immersive VE. *IEEE Computer Graphics & Applications*, vol. 18, pp. 9-13, 1998.
- [23] D. Bowman, J. Wineman, L. Hodges and D. Allison. The Educational Value of an Information-Rich Virtual Environment, *Presence: Teleoperators and Virtual Environments*, vol. 8, pp. 317-331, 1999.
- [24] D. Bowman, M. Setareh, M. Pinho, N. Ali, A. Kalita, Y. Lee, J. Lucas, M. Gracey, M. Kothapalli, Q. Zhu, A. Datey and P. Tumati. Virtual-SAP: An Immersive Tool for Visualizing the Response of Building Structures to Environmental Conditions, *Proceedings of IEEE Virtual Reality*, Los Angeles, pp. 243-250, 2003
- [25] B. Froehlich, Y. Kitamura and D. Bowman. Proceedings of "Beyond Wand and Glove Based Interaction", a workshop at *IEEE Virtual Reality 2004*. 2004.
- [26] D. Bowman, B. Froehlich, Y. Kitamura and W. Stuerzlinger. New Directions in 3D User Interfaces. Shaker Verlag, 2005.
- [27] Y. Kitamura, D. Bowman, B. Froehlich and W. Stuerzlinger. Proceedings of the IEEE Symposium on 3D User Interfaces. *IEEE Computer Society Press*, 2006.
- [28] I. Poupyrev, S. Weghorst, M. Billinghurst and T. Ichikawa. A Framework and Testbed for Studying Manipulation Techniques for Immersive VR, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 21-28, 1997



- [29] J. Gabbard, D. Hix and J. Swan. User-Centered Design and Evaluation of Virtual Environments, *IEEE Computer Graphics & Applications*, vol. 19, pp. 51-59, 1999.
- [30] J. Chen and D. Bowman. Effectiveness of Cloning Techniques for Architectural Virtual Environments, *Proceedings of IEEE Virtual Reality*, Alexandria, Virginia, 2006.
- [31] C. Wingrave, R. Tintner, B. Walker, D. Bowman and L. Hodges. Exploring Individual Differences in Raybased Selection: Strategies and Traits, *Proceedings of IEEE Virtual Reality, Bonn, Germany*, pp. 163-170, 2005
- [32] A. Olwal and S. Feiner. The Flexible Pointer: An Interaction Technique for Augmented and Virtual Reality, *Proceedings of ACM Symposium on User Interface Software and Technology*, pp. 81-82, 2003
- [33] R. Jacob, L. Deligiannidis and S. Morrison. A Software Model and Specification Language for Non-WIMP User Interfaces, *ACM Transactions on Computer-Human Interaction*, vol. 6, pp. 1-46, 1999.
- [34] P. Baudisch, E. Cutrell and G. Robertson. High-Density Cursor: A Visualization Technique that Helps Users Keep Track of Fast-Moving Mouse Cursors, *Proceedings of INTERACT conference, Zurich, Switzerland*, pp. 236-243, 2003
- [35] D. Vogel and R. Balakrishnan. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays, *Proceedings of ACM Symposium on User Interface Software and Technology*, Seattle, Washington, pp. 33-42, 2005.
- [36] F. Guimbretiere, M. Stone and T. Winograd. Fluid Interaction with High-resolution Wall-size Displays, *Proceedings of ACM Symposium on User Interface Software and Technology*, Orlando, FL, pp. 21-30 2001.
- [37] D. Bowman. Conceptual Design Space: Beyond Walk-through to Immersive Design, in *Designing Digital Space*, D. Bertol, Ed. New York: John Wiley & Sons, pp. 225-236, 1996
- [38] R. Bukowski and C. Sequin. Object Associations: A Simple and Practical Approach to Virtual 3D Manipulation, *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pp. 131-138, 1995.
- [39] Y. Kitamura, A. Yee and F. Kishino. Virtual Object Manipulation Using Dynamically Selected Constraints with Real-Time Collision Detection, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 173-181, 1996
- [40] J. Chen, D. Bowman, J. Lucas and C. Wingrave. Interfaces for Cloning in Immersive Virtual Environments, *Proceedings of Eurographics Symposium on Virtual Environments*, Grenoble, France, pp. 91-98, 2004.
- [41] J. Lucas. Design and Evaluation of 3D Multiple Object Selection Techniques in *Computer Science*, vol. M.S. Blacksburg, VA: Virginia Polytechnic Institute and State University, 2005.
- [42] B. Conner, S. Snibbe, K. Herndon, D. Robbins, R. Zeleznik and A. Van Dam. Three-Dimensional Widgets, *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pp. 183-188, 1992
- [43] M. Mine. ISAAC: A Meta-CAD System for Virtual Environments. *Computer-Aided Design*, vol. 29, pp. 547-553, 1997.
- [44] D. Mapes and J. Moshell. A Two-Handed Interface for Object Manipulation in Virtual Environments, *Presence: Teleoperators and Virtual Environments*, vol. 4, pp. 403-416, 1995.
- [45] M. Mine, F. Brooks and C. Sequin. Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction, *Proceedings of ACM SIGGRAPH*, pp. 19-26, 1997
- [46] J. Lucas, J. Kim and D. Bowman. Resizing beyond widgets: object resizing techniques for immersive virtual environments, *Proceedings of CHI 2005 Conference on Human Factors in Computing Systems*, Portland, OR, pp. 1601-1604, 2005
- [47] D. Bowman, C. North, J. Chen, N. Polys, P. Pyla and U. Yilmaz. Information-Rich Virtual Environments: Theory, Tools, and Research Agenda, *Proceedings of ACM Symposium on Virtual Reality Software and Technology*, Osaka, Japan, pp. 81-90, 2003
- [48] J. Chen, P. Pyla and D. Bowman. Testbed Evaluation of Navigation and Text Display Techniques in Information-Rich Virtual Environments, *Proceedings of IEEE Virtual Reality*, Chicago, 2004, pp. 181-188.
- [49] N. Polys, S. Kim, D. Bowman. Effects of Information Layout, Screen Size, and Field of View on User Performance in Information-Rich Virtual Environments, *Proceedings of ACM Symposium on Virtual Reality Software and Technology*, Monterey, California, pp. 46-55, 2005.
- [50] R. Stoakley, M. Conway and R. Pausch. Virtual Reality on a WIM: Interactive Worlds in Miniature, *Proceedings of CHI: Human Factors in Computing Systems*, pp. 265-272, 1995
- [51] R. Pausch, T. Burnette, D. Brockway and M. Weiblen. Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures, *Proceedings of ACM SIGGRAPH*, pp. 399-400 1995
- [52] J. LaViola, D. Acevedo, D. Keefe and R. Zeleznik. Hands-Free Multi-Scale Navigation in Virtual Environments, *Proceedings of ACM Symposium on Interactive 3D Graphics*, Research Triangle Park, North Carolina, pp. 9-15, 2001
- [53] C. Wingrave, Y. Haciahetoglu and D. Bowman. Overcoming World in Miniature Limitations by a Scaled and Scrolling WIM, *Proceedings of IEEE Symposium on 3D User Interfaces*, pp. 11-16, 2006
- [54] C. Wingrave and D. Bowman. CHASM: Bridging Description and Implementation of 3D Interfaces, *Proceedings of IEEE VR 2005 Workshop on New Directions in 3D User Interfaces*, Bonn, Germany, pp. 85-88, 2005



**Doug A. Bowman** He is a member of the ACM and the IEEE Computer Society. He is serving as the general chair of the 2007 IEEE Virtual Reality Conference. Doug A. Bowman received a B.S. in mathematics and computer science from Emory University (Atlanta, Georgia) in 1994, an M.S. in computer science from the Georgia Institute of Technology (Atlanta, Georgia) in 1997, and a Ph.D. in computer science, also from Georgia Tech, in 1999.

He is currently an Associate Professor in the Department of Computer Science at Virginia Polytechnic Institute and State University (Virginia Tech) in Blacksburg, Virginia. He is also affiliated with the Center for Human-Computer Interaction at Virginia Tech. He is the lead author of *3D User Interfaces: Theory and Practice?* (Boston, MA: Addison-Wesley, 2005), and has published more than 60 articles in peer-reviewed journals and conferences. His research interests include 3D user interfaces, 3D interaction techniques, and the benefits of immersion in virtual environments.