INTERACTION WITH VOLUMETRIC DISPLAYS

by

Tovi Grossman

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Interaction with Volumetric Displays

Ph.D. Thesis, 2008
Tovi Grossman
Graduate Department of Computer Science
University of Toronto

Abstract

For almost 50 years, researchers have been exploring the use of stereoscopic displays for visualizing and interacting with three-dimensional (3D) data. Unfortunately, a number of unfavorable qualitative properties have impeded the wide-spread adoption of traditional 3D displays. The volumetric display, a more recent class of 3D display to emerge, possesses unique features which potentially makes it more suitable for integration into workplace, classroom, and even home environments. In this dissertation we investigate volumetric displays as an interactive platform for 3D applications.

We identify the inherent affordances unique to volumetric displays, such as their true 3D display volume, 360° viewing angle, and enclosing surface. Identifying these properties exposes human factor issues which we investigate and interaction issues which we address. First, we evaluate the user's ability perceive imagery displayed by a volumetric display. In a formal experiment, we show that depth perception can be improved, in comparison to more traditional platforms. We then perform an experiment which evaluates users' ability to read text under 3D rotations, and present a new algorithm which optimizes text rotation when viewed my multiple users. Next, we investigate the user's ability to select 3D imagery within the display. Results show that the dimension defining the depth of the object can constrain user performance as much as or more than the other two dimensions of the target. This leads us to explore alternative methods of

selection which are less constraining to the user. We define a suite of new selection techniques, of which several are found to have significant benefits in comparison to techniques traditionally used in 3D user interfaces. Next, we describe our development of the first working interactive application, where a volumetric display is the sole device for input and display. The application presents a first glance at what the equivalent of today's graphical user interface might be on a volumetric display. We then develop a prototype application which allows multiple users to simultaneously interact with the volumetric display. We discuss and address the core issues related to providing such a collaborative user interface, and report feedback obtained from usage sessions and expert interviews.

Acknowledgements

This work would not have been possible without the guidance and supervision of Ravin Balakrishnan. I also thank my dissertation committee members, Ron Baecker, Mark Chignell, and Karan Singh, and additional Final Oral Examination committee members Khai Truong and Doug Bowman. In particular, I thank Mark Chignell for his provoking discussion as to the magical nature of Latin Squares, and Ron Baecker, for the thoroughness of his reading (he was even able to find a typo in a figure which I had been using in all of my job interview talks). I also thank the above committee members for being flexible in their schedules, so that meeting times could be agreed upon. I now believe that any graduate student that can coordinate the schedules of 5+ faculty members, without offending anyone in the process, is probably deserving of a PhD, regardless of the quality of their thesis.

I also thank those who have provided me with positive research experiences outside of University of Toronto. At Alias|Wavefront - Gord Kurtenbach, George Fitzmaurice, Azam Khan, and Bill Buxton; and at Microsoft Research - Ken Hinckley, Patrick Baudisch, and Maneesh Agrawala. In particular, the research group at Alias|Wavefront was an amazing introduction for me into the worlds of both research and Human-Computer Interaction. I can only imagine what trouble I would have gotten myself into by now if I hadn't stumbled upon that lab and spent the summer of 2000 there as a volunteer research assistant.

I must also thank all of my friends and family, who have supported me and helped shape me into the person that I am today. Of course, I must thank in particular my wife Anne, who has always been there for me, and who has not only supported me in my research career, but has dedicated countless hours to proof reading papers the night before submission deadlines. She even managed to stay awake while doing so – for most of them, that is. I should also thank my son, Chazz, who never ceases to inspire and motivate me.

From baseball, I have learned that a perfect team, which is destined for success, is one in which each individual cares equally about their own performance, their team's performance, but also each of their teammates, regardless of their performance. To be part of such a team is truly an incredible experience, creating lifelong memories and friendships. In my baseball career, I have been lucky enough to be part of such teams, and so I thank my teammates, and especially the coaches, from which these experiences stem: Geoff Cullen, Rob Scicluna, Andrew Green, Ian Bugless, Dan Lang, Eugene Ristuccia, and Tommy Adams. I can only hope to be fortunate enough to create, and be a part of, such teams in my future career endeavors.

Electronic Archives and Videos

Electronic copies of this thesis are archived at:

http://www.dgp.toronto.edu/~tovi/thesis

https://tspace.library.utoronto.ca/handle/1807/9944

These archives include the thesis document, as well as the following video figures:

Video Figure 6-1
The design and evaluation of selection techniques for 3D volumetric displays.

Video Figure 7-1
Multi-finger gestural interaction with 3D volumetric displays.

Video Figure 8-1
Collaborative interaction with volumetric displays.

Association for Computing Machinery Copyright Notice

Several sections of this document have previously appeared in ACM publications. Permission from ACM has been granted for these works to appear in this thesis. Where appropriate, each chapter is marked with references to the relevant publication, each of which is listed below:

Chapter 3
[Grossman and Balakrishnan 2006a] Grossman, T. and R. Balakrishnan (2006a). An evaluation of depth perception on volumetric displays. *Proceedings of the Working Conference on Advanced Visual Interfaces*, 193-200.

Chapter 4
[Grossman et al. 2007] Grossman, T., D. Wigdor and R. Balakrishnan (2007). Exploring and Reducing the Effects of Orientation on Text Readability in Volumetric Displays. *ACM CHI Conference on Human Factors in Computing Systems*, 483-492.

Chapter 5
[Grossman and Balakrishnan 2004] Grossman, T. and R. Balakrishnan (2004). Pointing at trivariate targets in 3D environments. *ACM CHI Conference on Human Factors in Computing Systems*, 447-454.

Chapter 6
[Grossman and Balakrishnan 2006b] Grossman, T. and R. Balakrishnan (2006b). The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. *ACM UIST Symposium on User Interface Software and Technology*, 3-12.

Chapter 7
[Grossman et al. 2004] Grossman, T., D. Wigdor and R. Balakrishnan (2004). Multi finger gestural interaction with 3D volumetric displays. *ACM UIST Symposium on User Interface Software and Technology*, 61-70.

Chapter 8
[Grossman and Balakrishnan 2008] Grossman, T. and R. Balakrishnan (2008 - in press). Collaborative interaction with volumetric displays. *ACM CHI Conference on Human Factors in Computing Systems*, 383-392.

Table of Contents

1. Introduction

"All right everyone, line up alphabetically according to your height."

-Casey Stengel

In 1968, Sutherland implemented the first system which coupled a tracked head-mounted display with real-time three-dimensional (3D) computer graphics [Sutherland 1968]. The system introduced the idea of displaying 3D imagery in a way which fully replicates the physical experiences we have with our 3D environment. This is in contrast to even today's traditional desktop workstations, where 3D data can be visualized, but the imagery is only perceived as a 2D representation. One cannot deny that such traditional desktop setups have allowed users to successfully carry out a wide variety of 3D tasks, such as 3D modeling, animation, planning, and visualization applications. However, in providing the user with a more realistic 3D representation of the data which they work with, there is a potential to improve the overall experience when carrying out virtual 3D tasks [Arthur et al. 1993].

Recognizing this potential, an abundance of research in 3D display systems has followed Sutherland's pioneering work. Immersive head-mounted displays have reduced in size and weight, and increased in quality [Neale 1998]. Less immersive systems have also been explored, such as fish tank virtual reality, where shutter glasses are synchronized with a traditional CRT monitor to produce stereoscopic imagery [Ware et al. 1993]. While both classes of display provide high quality 3D imagery, a number of qualitative issues have prevented their adoption.

One critical factor is that users are required to wear glasses, head trackers, and other specialized hardware. This would be particularly undesirable for high-volume users; a car designer, for example, would not want to wear a head mounted display for an entire work-day. Even a casual user may be resistant to the idea of wearing hardware. Requiring supplementary hardware also reduces the possibility of these displays being deployed in shared or public locations, as the user cannot simply walk up and use them. Furthermore, immersive 3D technologies can cause the user to lose the context of their surrounding environment and collaborators [Arthur et al. 1993].

The second issue is that typical 3D displays actually present imagery on a 2D plane. As a result, there is a discrepancy in the depth cues which they provide to the user. This discrepancy can cause nausea, dizziness, and headaches [McCauley and Sharkey 1992, Shibata et al. 2005].

Since the development of Sutherland's first head mounted system, numerous classes of 3D displays have emerged. However, almost all suffer from one or both of the above issues. An exception, which is the focus of this dissertation, is the volumetric display.

1.1 Volumetric Displays

Volumetric displays [Blundell et al. 1993, Ebert et al. 1999, Favalora 2005, Langhans et al. 1998] generate true volumetric 3D images by actually illuminating points in 3D space. As such, viewing imagery on volumetric displays is akin to viewing physical objects in the real world. Viewers can use their inherent physiological mechanisms for depth perception to gain a rich understanding of the virtual 3D scene. These displays typically have a 360° field of view, and the user does not have to wear hardware such as shutter glasses or head-trackers. As such, they are a promising alternative to traditional display systems for viewing in 3D (Figure 1-1).

We see volumetric displays as having the capability to change the way we carry out virtual 3D tasks – from common tasks such as online browsing of commercial products, to specialized tasks, such as inspecting the results of a medical scan. Although these displays are now commercially available (e.g., www.actuality-systems.com), current applications tend to use them as a non-interactive output-only display device, much like one would use a printer. In order to fully leverage the unique features of these displays, it would likely be desirable if one could directly interact with and manipulate the displayed 3D data. For example, imagine a doctor not just being able to view a medical scan, but to be able to manipulate the imagery, by interactively cutting slices away or peeling off layers, to reveal other areas.

Figure 1-1. A 3D volumetric display. Voxels illuminated in true 3D space facilitate 3D viewing from anywhere around the display without wearing special glasses.

1.2 Display Properties and Interaction Issues

Before such applications for volumetric displays can be developed, it is essential to conduct a thorough exploration of the issues involved in making them an interactive platform. A fair amount of work in the field concerning 3D interaction exists, largely in the virtual reality literature [Bowman and Hodges 1997, Mine 1995a]. However the new properties which volumetric displays possess merit further investigation.

Identifying these important properties relevant to interacting with volumetric displays is a challenge in itself, as the displays are only in an early stage of development, and few interactive usage scenarios have been proposed. To aid in this process we have made use of a working volumetric display, made by Actuality Systems [Favalora et al. 2002]. This display has helped us explore the affordances of volumetric displays, along with allowing us to perform informal usage observations of users viewing static imagery on the display. This, in combination with reference to previous exploratory studies on user interfaces for volumetric displays [Balakrishnan et al. 2001] has allowed us to identify important properties of volumetric displays which have implications to the development of user interfaces and interaction techniques. Below, we discuss these properties, along with the relevant research questions and hypotheses which arise from them.

1.2.1 Autostereoscopic Display of Imagery

Three-dimensional display technologies, such as immersive VR systems [Buxton and Fitzmaurice 1998, Neale 1998, Sutherland 1968], require users to wear cumbersome head mounted displays, while the less intrusive fish-tank VR displays [Ware et al. 1993], still require users to wear special glasses. Furthermore, for these systems to provide head coupled viewing, where a user's viewpoints is based on the user's head position, special hardware must be used to track the location of the user's head. Volumetric displays, on the other hand, illuminate points in true 3D space, so special hardware, such as shutter glasses or head trackers, is not required. Single users can thus work with volumetric displays while maintaining the context of their physical surroundings, while multiple users can view the scene simultaneously while maintaining the context of their collaborators, possibly allowing for richer multi-user experiences.

An important implication of this property is that applications using the volumetric display can be "walk-up and use", without an associated start-up cost of putting on glasses, head-gear, or tracking devices. This may make the display particularly suitable for shared or public spaces, such as a central location of a design studio, or at the back of the classroom – since at any time, anyone could approach the display to view or interact with the displayed imagery. For the design of such applications, it may be desirable to not introduce external input devices, so that the volumetric display is the sole platform for all display and input, and completely free of any supplementary hardware that the user needs to find, pick up, and put down. This introduces the following research question:

> ***Research Question: What interfaces and interaction techniques will allow users to directly interact with volumetric display applications, without the need for any supplementary input devices?***

We believe that a user interface which relies on gestural input could be developed to allow for direct interaction with the volumetric display, using the surface as the platform for interaction. In Chapter 7 we will explore how such a system could be implemented.

1.2.2 Consistent Depth Information

Another limitation of traditional stereoscopic displays is that they create a conflict between the two mechanisms that enable human stereoscopic vision: accommodation and convergence (Figure 1-2). The resulting ambiguous depth information can cause users to experience symptoms of asthenopia, such as nausea, dizziness and eye fatigue [McCauley and Sharkey 1992, Shibata et al. 2005]. With volumetric displays, since images are produced in true 3D space, the convergence and accommodation cues are consistent, eliminating the symptoms of asthenopia.

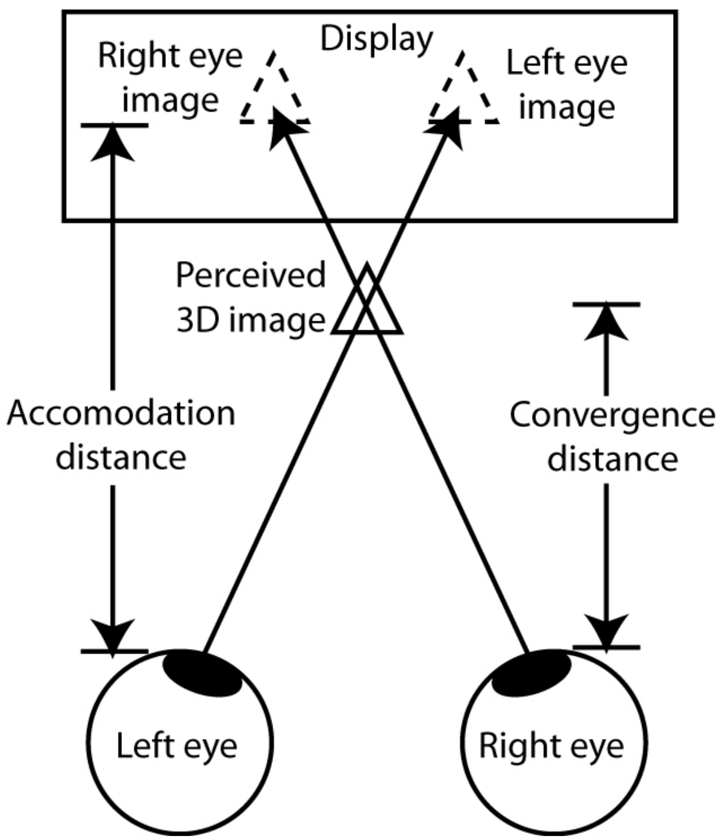

Figure 1-2. With most stereoscopic displays, there is a discrepancy between the accommodation and convergence depth cues. The accommodation distance is the distance between the user's viewpoint and the display plane. In a stereoscopic display, this distance remains fixed. The convergence distance is the distance between the user's viewpoint and the point where the viewing axes between the two eyes intersect. This distance is dynamic, and determines where the viewer perceives a virtual object to exist.

An implication of this is that it becomes reasonable to expect a user to work with a volumetric display on regular basis, since negative side-effects to working with the system are not to be expected. Another implication is that viewing imagery on the volumetric display is equivalent to viewing physical objects in the real world. This raises a human factors question related to the user's ability to accurately perceive virtual 3D data, in comparison to more traditional 3D display technologies:

> ***Research Question: Do volumetric displays possess measurable viewing benefits in comparison to more traditional 3D display devices?***
> *Our belief is that because volumetric displays project imagery in true 3D space, and thus provide the user with consistent depth cues, we should be able to find improvements in depth perception. We study this potential effect in Chapter 3.*

The true 3D nature of volumetric displays also raises an important question relating to users' ability to interact with imagery within the display, which will require pointing to the corresponding objects in 3D:

> ***Research Question: What factors affect direct pointing in volumetric displays, and can the task be understood with traditional models of pointing behavior?***
> *We expect pointing within volumetric displays, which is a 3D task, will be affected by numerous factors: all three dimensions of the target, the angle of movement towards the target, and the distance required to travel. Our hope is that the traditional 1D and 2D pointing models can be adapted to incorporate these additional factors. We investigate this topic in detail in Chapter 5.*

1.2.3 Full 360° viewing

Another interesting property of volumetric displays is that the imagery can be viewed from any angle around the display. This is because the imagery consists of points of lights being illuminated in true 3D space. This is in contrast to other forms of autostereoscopic displays [Dodgson 2005], which can only be viewed from certain angles, and to fish tank virtual reality platforms [Ware et al. 1993], for which the user's viewpoint must be in front of a planar display surface. This property has some important implications to interface design for volumetric displays.

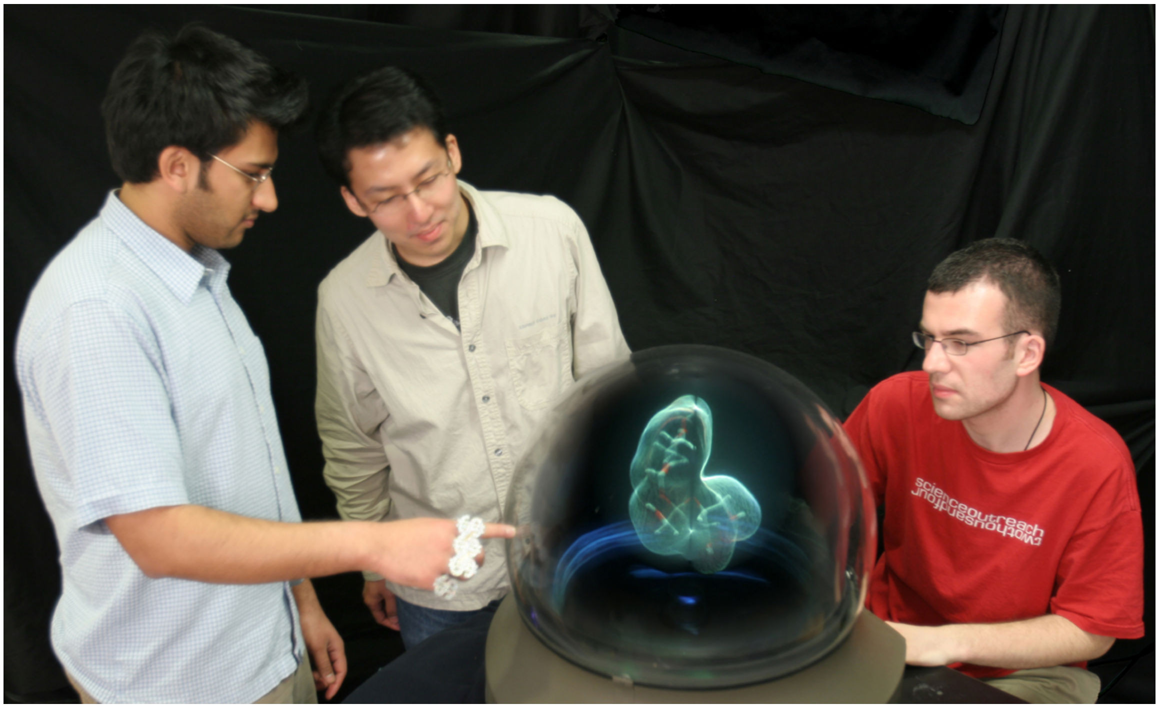

Figure 1-3. Multiple users can view and interact with imagery in a volumetric display from different viewpoints.

First, applications should be made “omnidirectional” – a user should have the freedom to easily interact with the system from anywhere around the display. This introduces interesting interaction challenges: elements of the user interface must be accessible from all locations, and application data must be viewable from all locations:

> ***Research Question: What challenges are introduced as a result of a user’s ability to view imagery from any angle around the display?***
> *A number of challenges will be introduced. Most notably, text will become difficult to read as it will appear at various orientations. Furthermore, new interaction challenges will arise since users will require access to the user interface regardless of their viewing position. We investigate these issues in Chapters 4 and 8.*

The other implication of the 360° viewing angle is that the display becomes particularly suitable for collaborative usage. Multiple users can simultaneously view and potentially interact with the displayed imagery from anywhere around the display (Figure 1-3). As such it is important to fully investigate the interaction issues which arise when designing applications for collaborative interaction:

> ***Research Question: What issues are introduced when multiple users interact with volumetric displays, and what new interaction techniques can be developed to address those challenges?***

We believe that if the volumetric display is used a collaborative platform, then the main issues will be that each user will have a unique viewpoint of the data, and users will need to access the user interface from their respective locations. Interaction techniques will need to be developed to overcome these potential challenges. We explore such issues in Chapter 8.

1.2.4 Enclosed Display Volume

In virtual reality systems, imagery is suspended in mid-air, and there is no perceived physical barrier between the user and the image. Users can navigate towards objects of interest, and directly grab and manipulate the objects once they are within arm’s reach [Mine 1995a]. A current property of volumetric displays is the existence of a physical enclosure of the display, which acts as a barrier between the user and the displayed imagery.

The obvious implication of this property is that users cannot directly reach in and grab objects, even when they are within arm’s reach. This presents a new and interesting challenge which warrants the exploration of new interaction paradigms, especially for selection:

Research Question: What techniques will make the task of selecting objects within volumetric displays easier?
We predict that a direct extension of the traditional GUI cursor into 3D will be problematic, as it would require precise positioning in 3D. We believe that selection techniques which do not require 3D positioning should be developed, and will provide better performance. In Chapters 5 and 6 we explore alternatives to direct grabbing for the selection of virtual imagery.

Another implication is that the display surface provides a potential platform for user input. In our informal observations of users viewing static imagery, it was common to see users trying to manipulate the imagery by scrubbing on the surface of display. This implies that the display enclosure affords interaction with its surface. In Chapter 7, we explore how the enclosure can be made use of, by using it as an interactive touch sensitive surface.

1.3 Display Limitations

The unique properties outlined above make volumetric displays an interesting platform for interactive 3D applications. However, the displays do have a number of limitations, which make them most appropriate for certain types of applications and tasks. For one, they only provide an outside in, or “God’s eye view” of the 3D data, and cannot provide an immersive, within world, experience. Related to this, they only provide a limited viewing area, and unlike virtual reality environments, cannot display infinitely large scenes. A final limitation, with the current generation of displays, is that imagery cannot be rendered opaque, due to the technological implementations. In Section 9.4, we will revisit some of these limitations, and discuss how our work could be adapted, if these limitations are ever addressed by the technology.

1.4 Hardware Platform

The experiments, user studies, and prototypes which we will discuss in the remaining chapters of this thesis were all implemented on the same volumetric display: Actuality System’s Perspecta Spatial 3D Display. We now give a brief description of the system. For a more through description we refer the reader to the work by Favalora et al. [2002].

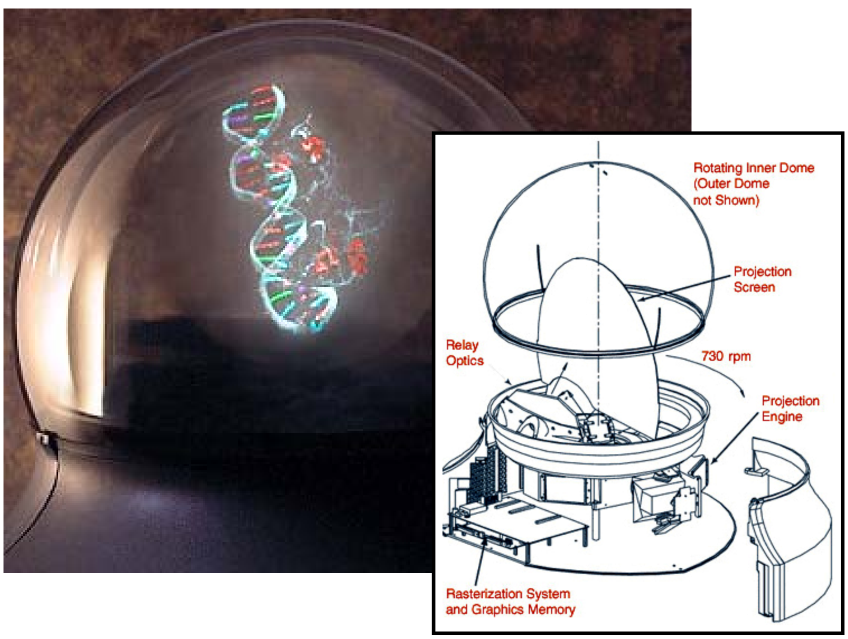

Figure 1-4. The Perspecta Spatial 3D Display, developed by Actuality Systems. Inset: Schematic of the display hardware and mechanics.

The system is a swept volumetric display with a voxel resolution of 768 x 768 x 198. The operation of the display is based on the theory of human persistence of vision, which is capable of integrating multiple slices into a single volume-filling 3D image. A 10 inch circular diffuse projection screen rotates inside the volume at 730rpm. A stationary high-speed projection engine projects approximately 5000 frames per second onto the screen, via a series of 3 mirrors. The mirrors which relay the imagery are placed to ensure accurate focus regardless of the screen's angle. This sequence of 2D images is perceived as a single 3D volume due to persistence of vision.

Each of these 2D images is a cross-section of a 3D dataset. The resulting 3D imagery can be seen from any location around the display. Because the screen is essentially a Lambertian diffuser, each voxel appears to emit omnidirectionally. As a result, the imagery is perceived as translucent and hidden-surface removal is not possible without knowledge of the viewer's position. Since the display provides imagery in true 3D space, no headwear is required, and accommodation and convergence cues are correct.

A three-chip digital light processing (DLP) projector is used to create the 3-bit color 2D images at approximately 5 kHz. A prism is used to split the light into its R, G, and B, components. Each of these components passes through a single digital micromirror device (DMD) chip. The DMD chips consist of a 1024x768 array of micromechanical mirrors, corresponding to the pixel resolution of the 2D slices (although only the central 768x768 pixels are used). The MEMS are used to either let light pass through, or to reflect the light away. A color mixing prism is used to combine the three components with 1-bit depth each, with a resulting 3 bit pixel being projected onto the screen.

1.5 Outline and Contributions

The first contribution of the current work is the identification of the inherent properties of volumetric displays (Section 1.2). It is these properties which raise new human factor questions and interaction challenges, which we explore in detail in the remaining chapters of this thesis. Because volumetric displays are a relatively new platform, research in this area is limited (Section 2.6.5). As a result, we will be laying the groundwork for future applications and research with 3D volumetric displays to build upon. This work will build upon numerous related research areas, which we outline in Chapter 2.

One inherent property which we identified in Section 1.2.2 is the true 3D nature of the volumetric display's imagery. In Chapter 3 we investigate the resulting impact on the user's ability to perceive the displayed imagery. We report the results of a formal experiment, where users perform three depth perception tasks on a variety of display configurations. This study contributes an empirical comparison of volumetric displays to more traditional 3D display technologies. An important finding is that volumetric displays do improve low-level depth perception, meaning they have the potential to improve a user's ability to carry out 3D tasks.

In Chapter 4, we continue our investigation into the perceptual issues associated with the display, this time focusing on the user's ability to read text under the rotations which are inherent when viewing imagery on the volumetric display. In an initial experiment, we evaluate the effect of pitch and yaw rotations on reading times. We contribute empirical data, and propose techniques which are shown to improve reading times, such as using unambiguous font sets. Based on the results of this study, we introduce a new algorithm which optimizes the orientation of text when being viewed by multiple users. We show that this technique can significantly reduce average group reading times in a formal experiment.

In Chapter 5, we study the human's capability to perform selections using a status-quo 3D cursor, overcoming the enclosure which separates input and display. We identify the relevant factors which affect the user's performance in this fundamental task, and propose a new predictive model which is validated through a formal experiment. The study has important implications to the design of user interfaces and selection techniques for volumetric displays. Another contribution of the work presented in this chapter is that our model is the first validated extension of Fitts' law to the third dimension. This result is independent of the display device, and thus applicable outside the scope of volumetric display research, and to the general area of 3D user interfaces.

Motivated by the results obtained in Chapter 5, we design and evaluate a suite of new selection techniques for volumetric displays, presented in Chapter 6. In an initial study we find that a ray casting metaphor can improve upon the performance of a 3D cursor. Based on this result, we introduce four new selection techniques, which address the ray cursor's ambiguity problem when multiple objects fall along the ray. In a formal experiment we found that our new techniques were superior to the baseline 3D point cursor. In particular, the *depth ray* provided the best overall results in terms of speed, accuracy and device footprint. Another significant contribution of this work is the proposed solution to the ray cursor's ambiguity problem, which has been an open problem in general 3D user interface research. Our solutions can be generalized to selection techniques for other 3D platforms, such as virtual reality and fish tank virtual environments.

In Chapter 7, we build upon these lower level studies, and we explore how interaction techniques, such as selection, can be integrated into high-level user interfaces for volumetric displays. The contribution is a first glance at what the equivalents of today's graphical user interfaces might be on a volumetric display. To ensure its appropriateness, we carry out this design exercise in consideration of the unique properties of the display. The investigation is achieved through the development of an interactive 3D geometric model building application; however the interactions which are developed can be generalized to any volumetric display application. The result is the first known interactive system which uses the volumetric display as the main platform for both input and output. We explore a very direct style of interaction, where the user interacts with the virtual data using direct finger manipulations on and around the enclosure surrounding the displayed 3D volumetric image. Interface elements such as menus, file browsers, and widgets are implemented, and important 3D functionality is supported, such as rotating and scaling imagery.

As discussed earlier in this chapter, a number of the properties of volumetric displays make them particularly suitable for multiple users. In Chapter 8, we discuss some of the key interaction challenges which are raised when designing volumetric display applications for collaborative use. Example issues are allowing users to indicate areas of interest to one another, allowing users who are positioned in different areas around the display to share their unique viewpoints with one another, and supporting both shared and personalized working volumes. We explore potential solutions to these challenges by implementing a prototype application which allows multiple users to view and markup a 3D model. The application serves as a platform for a thorough exploration of interaction techniques for collaborative applications for volumetric displays.

As discussed in Section 1.4, our studies and prototypes were implemented on a single volumetric display. However, our results can be generalized to other implementations of volumetric displays. Such displays may vary in their technological implementation, but more relevant to our work, they may also vary in their physical affordances, such as size and shape. In Chapter 9, we discuss how our results would be affected if the volumetric display form did vary. Furthermore, many of our results are applicable to displays with properties that diverge from the four central properties of volumetric displays, which we have discussed in this chapter. In Chapter 9, we will also consider how our results could be applied to other types of displays. We will also summarize the scope of the work presented in this thesis, list the main contributions, and discuss possible future lines of work in Chapter 9.

2. Background Literature

"Ability is the art of getting credit for all the home runs somebody else hits."

-Casey Stengel

2.1 Introduction

In Section 1.2, we discussed the new affordances presented by volumetric displays. These affordances raise new human factors issues and interaction challenges. Once these new topics are better understood, applications, and the user interfaces and interaction techniques which they are made up of, can be appropriately designed. In this chapter, we review the previous research which will help guide our investigation into making volumetric displays an interactive platform.

The chapter will be structured as follows. In Section 2.2 we present an overview of existing 3D display technologies, followed by a description of existing volumetric display technology in Section 2.3. Section 2.4 looks at methods for evaluating the viewing experience in these various 3D displays, which can be extended to the study of volumetric displays. In Section 2.5, we provide an overview of the input devices used for interacting with 3D environments. In Section 2.6, we look at some of the basic interaction techniques existing in the virtual reality and 3D interaction communities, which will provide insights into the development of interaction techniques for volumetric displays. In Section 2.7, we look at work which integrates these interaction techniques into 3D applications. Section 2.8 focuses on collaborative applications, which will aid in the development of interaction techniques for volumetric displays, under a multiple user scenario. Because there is such a large collection of related research, we focus our literature review on the most relevant and initial works where the concepts appeared.

2.2 Three-Dimensional Display Technology

In this section we provide a background on the types of display technology which have been used to present three-dimensional data. These display techniques all utilize various depth cues which have been identified in psychological research on human perception. We now discuss the display techniques, and the associated cues which these displays provide.

2.2.1 Perspective Displays

A basic method for displaying three-dimensional environments, which is commonly used in 3D graphics, is to use a standard workstation monitor, using a perspective projection. Perspective displays utilize perspective and relative size cues, in which objects further away, produce smaller retinal images than closer objects. Such displays are commonly exploited in 3D graphics [Foley et al. 1990]. Brooks [1988] notes that perspective cues are particularly effective when parallel lines are displayed within the scene.

Another cue which is utilized in perspective displays is motion parallax, also referred to as the kinetic depth effect. With this cue, a sensation of depth is produced when an object moves in space relative to an observer. For this reason, interfaces should provide a mechanism for moving the viewpoint in a 3D scene [Brooks 1988, Lipscomb 1981]

By utilizing these two powerful depth cues, perspective displays are effective tools for viewing and interacting with three-dimensional data, and are still the most commonly employed display type for tasks such as 3D graphics design and animation.

2.2.2 Non-Immersive Stereoscopic Displays

An important depth cue which cannot be provided by a standard workstation monitor is stereopsis. This depth cue is produced from the binocular disparity when viewing 3D objects in natural environments. It can be particularly strong when the perceived objects are close to the viewer [Yeh 1993]. To provide stereopsis, a system simply needs to present different images for the left and right eyes, to produce the required binocular disparity. A fused 3D image appears at the point of convergence between the two images.

There are a number of display technologies which can provide stereopsis on a flat screen [Arditi 1986, McAllister 1993]. Common examples of stereoscopic displays are polarized glasses which passively block certain light from each eye and liquid-crystal time-multiplexed shuttering glasses which actively block light from each eye [Lipton 1985]. Autostereoscopic displays are another class of stereoscopic displays, which are glasses free systems that project multiple images for each eye in specific locations in space [Dodgson 2005].

Figure 2-1. Fish tank displays combine stereoscopic and head tracking technology to create a viewing volume roughly equivalent to the inside of the monitor. Image taken from Ware and Franck [1996].

2.2.3 Semi Immersive Fish Tank Displays

The display technologies discussed so far are generally considered to be non-immersive, as there is a clear distinction between the physical and virtual environments. Fish tank displays increase the level of immersion by coupling the viewpoint of the virtual scene, with the physical location of the user's head, so that the appropriate perspective is always presented [Diamond et al. 1982a, Fisher 1982, McKenna 1992b, Sollenberger and Milgram 1991, Sollenberger and Milgram 1993, Ware et al. 1993] (Figure 2-1). Generally the user's head position is tracked, and the location of the user's eyes is estimated by offsetting them by a constant distance from the user's head [Arthur et al. 1993]. The viewpoint of the virtual scene is then updated appropriately. The virtual scene can be perceived monocularly, coupled to a single eye position, or binocularly, if combined with one of the stereoscopic technologies discussed above. Because the scene is still projected on a 2D display, fish tank displays are considered to be semi-immersive. Generally the viewing volume of these displays is roughly equivalent to the inside of the monitor, so viewing them is similar to looking into a fish tank. The head coupled viewpoint with fish tank displays provides motion parallax cues, without requiring users to explicitly manipulate the camera position.

2.2.4 Head-Mounted Displays

More immersive forms of displays, typically associated with the field of virtual reality, which provide stereopsis and motion parallax through head coupled viewpoints, have also been developed. A common form is a head mounted display, which use some sort of helmet or goggles to provide a stereo pair of displays directly in front of each eye [Buxton and Fitzmaurice 1998, Neale 1998, Sutherland 1968] (Figure 2-2). Such displays provide the widest possible field of view at high quality. Head mounted displays can provide the user with a realistic 3D experience, but a significant problem is that the user's eyes are covered by the display. The result is a lost context of the user's physical surrounding and possible collaborators [Arthur et al. 1993, Buxton and Fitzmaurice 1998].

Figure 2-2. A head mounted display creates an immersive virtual reality by covering the user's eyes. Image taken from Buxton and Fitzmaurice [1998].

A solution to this problem is to allow users to see the physical world, such as their hands, tools, and other people, while wearing the display. This can be accomplished by mounting video cameras onto the displays, and incorporating their images into the virtual world which the user sees [Azuma and Bishop 1994, State et al. 1996, Yoo and Olano 1993]. The cameras act as an extra set of eyes for the user, projecting a view of the physical world onto the computer generated view of the virtual world. This is similar to providing the user with a heads up display, using an optical see-though display. This approach is generally known as augmented reality, as the systems augments the physical world with additional virtual imagery [Feiner et al. 1993b].

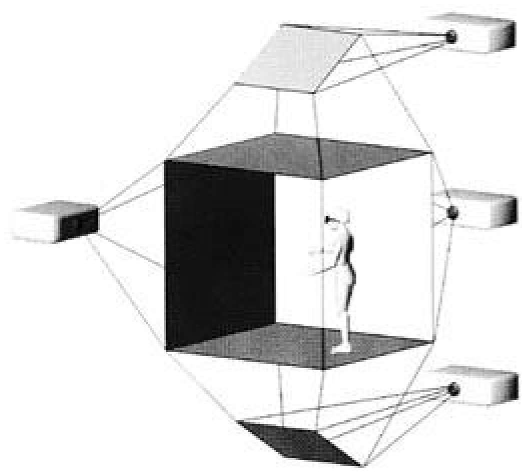

Figure 2-3. Schematic of a Cave system. Tiled rear projected images appear on multiple faces of the room. Figure taken from Cruz-Neira et al. [1992].

2.2.5 Cave

Another interesting form of an immersive 3D display is known as a Cave [Cruz-Neira et al. 1992]. A Cave is room where the walls, floor, and ceiling, act as the display (Figure 2-3). To provide a seamless view of the virtual scene, each of the displays must be tiled together at the edges where they intersect. The projected surfaces of the Cave are typically in stereo, with the user wearing a pair of glasses. Furthermore, the head is tracked so that the appropriate perspective is always presented to the user. A benefit of Caves over head mounted displays is that users can still see and have a maintained context of their physical surroundings. A number of other systems similar to Caves have also been developed, which vary in size, and also the number of surfaces which are projected onto. One example is large scale projection displays, such as the ImmersaDesk, which are stereoscopic but are only displayed on a single surface [Czernuszenko et al. 1997]. Small format caves have also been built, such as the Cubby system, where three head tracked back projection screens form a cubic workspace in front of the user [Djajadiningrat et al. 1997].

2.2.6 Chameleon

A less common technology for presenting 3D environments is the Chameleon system [Fitzmaurice 1993]. With the Chameleon system, perspective images of three-dimensional scenes appear on a small display held in the palm of the hand. As with fish tank and head mounted displays the perspective view is continuously updated. However the perspective viewpoint is determined by the position of the display, rather than being coupled with the head of the user.

The Chameleon approach is almost like using a magnifying glass which looks into a virtual scene, rather than the physical world. While the display does not provide the full immersive feeling or viewing angle of a cave or head mounted display, users can easily browse the scene by moving the lightweight display.

There have been other implementations of Chameleon displays. The boom chameleon [Tsang et al. 2002] consists of a flat-panel display mounted on a tracked mechanical boom. The armature of the boom is carefully balanced to allow the display to float weightlessly in space. The display acts as a physical window into 3D virtual environments, through which a one-to-one mapping between real and virtual space is preserved.

Chameleon systems can also have the ability to support augmented reality. Fitzmaurice [1993] showed how location tracking could also be used to sense the context of its environment. For example, bringing the display close to a map could give additional information about the region that it was close to. Rekimoto [1996] augmented Chameleon-like devices further by adding video cameras, allowing computer-generated information to be superimposed over a view of the physical world.

2.2.7 Summary

We have presented a variety of display technologies for three-dimensional data. The displays provide different levels of immersion to the users ranging from non immersive perspective displays to fully immersive head mounted displays. Each display has its own unique properties which afford different methods of interaction. In the next section, we will discuss various volumetric display technologies. By examining and comparing the properties of volumetric displays to the technologies discussed in this section, we will gain an understanding of what interaction techniques will transfer well to volumetric displays, and what interaction techniques will require modifications or enhancements.

2.3 Volumetric Displays

Unlike the other display technologies previously discussed volumetric displays present imagery in true 3D space. The viewing space is divided up into 3D pixels, called voxels, which illuminate visible light from the regions which they appear. Because light is emitted in true 3D space, depth cues such as motion parallax and stereopsis will always be present without the need for supplementary hardware. From a technological stand point, there are some interesting differences between the various implementations of volumetric displays. While they all generate true 3D images, the underlying technology of the various displays vary greatly in both concept and design. From an interaction standpoint, however, these differences will not be so critical, as all of these displays pose the same qualities which we have discussed in Section 1.2. We now provide an overview of the basic classes of 3D volumetric displays.

2.3.1 Oscillating Swept Volumetric Displays

A common method for presenting volumetric imagery is to use a two-dimensional image which periodically sweeps out a three-dimensional volume of space. When the space is swept out cyclically at a frequency higher than what the eye can resolve, a spatial image is formed through the persistence of vision. We now discuss the class of displays in which volumes are swept out by oscillating screens.

An early implementation uses a vibrating CRT where the images which output to the screen are synchronized within each oscillation cycle, so that each image appears at the appropriate position of the tube [Blundell et al. 1993]. Because of the high mass of the tube, it would be difficult to obtain the desirable rapid acceleration and deceleration at each end of its oscillation.

A more practical suggestion is to have the phosphorous screen within the CRT oscillate rather than the entire CRT itself [Rawson 1969]. The screen is mounted within a vacuum tube behind a transparent viewing globe, and oscillates with a piston like motion. An electron gun illuminates the screen from the rear (Figure 2-4).

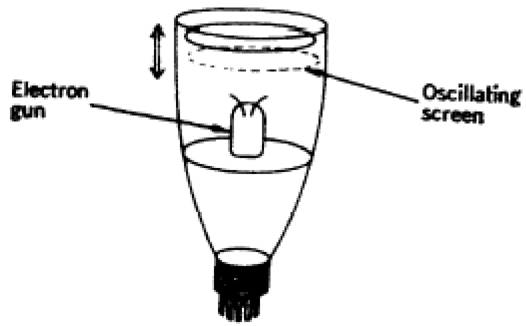

Figure 2-4. An oscillating phosphorous screen is illuminated by an electron gun. Figure taken from Langhans et al. [1998].

A variation of the oscillating screen, is to have an oscillating mirror from which a CRT image is viewed [Langhans et al. 1998]. The mirror moves such that the visual path length is swept periodically, and the CRT image apparently sweeps out the volume. The drawback of this scheme is that the viewing angle is limited to the reflective surface coverage.

Because of the inertial forces required for oscillating screens, this class of displays generally has problems with vibration and noise. The accelerations involved also limit the oscillation frequency, causing considerable flicker of the displayed images. A more promising approach is a rotating swept display, which we now discuss.

2.3.2 Rotating Swept Volumetric Displays

One of the earliest implementations of rotating swept volumetric displays was developed by the ITT Laboratories in 1960, which consisted of a cathode ray tube whose light is optically transferred to a translucent rotating screen within a glass cylinder [ITT-Laboratories 1960]. The motor driven panel turned at approximately 20 revolutions per second. Another implementation was presented in 1963, where a phosphor-coated screen rotated in a vacuum, with a controlled electron beam striking its surface. The 3D images were presented by having the electron beam project on the screen as it passes through the desired location of space [Ketchpel 1963].

The rotating screens can also be actively emissive, instead of having imagery projected onto them. Berlin proposed using a 2D matrix of light emitting diodes (LEDs) with the light electronics rotating to sweep out a 3D volume [Budinger 1984] (Figure 2-5). In this implementation, the display resolution is a function of the number and density of LEDs, the speed of rotation, and the rate at which LEDS can be pulsed.

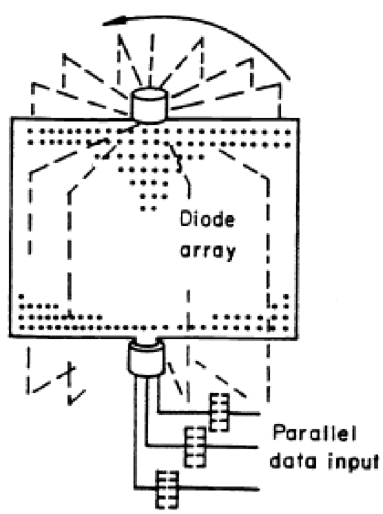

Figure 2-5. Rotating volumetric display using an emissive panel of LEDs. Figure taken from Langhans et al. [1998].

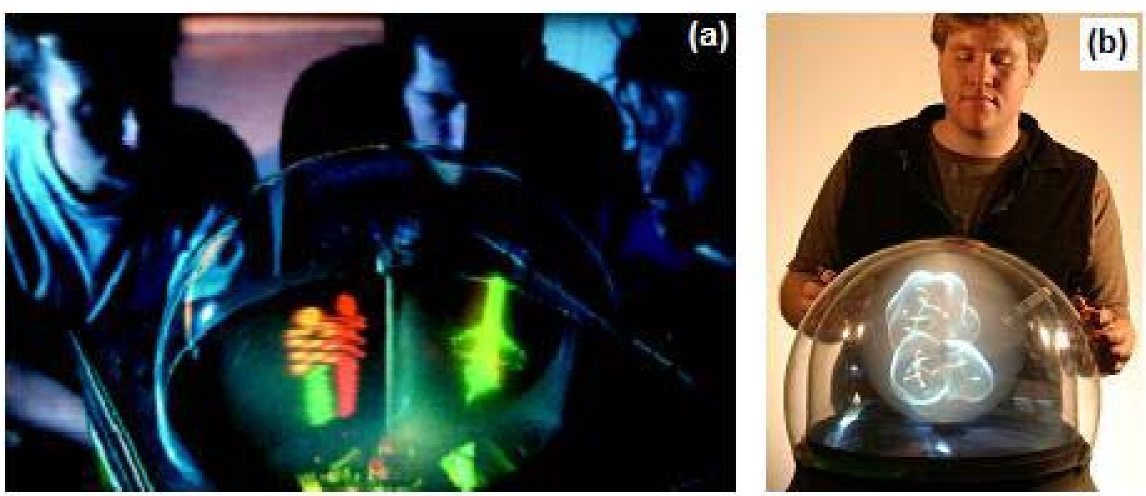

Figure 2-6. High-end rotating volumetric displays. (a) Felix: www.felix3d.com. Image taken from Balakrishnan et al. [2001]. (b) Perspecta: www.actuality.com.

The above rotating displays are swept out by flat surfaces. However in another class of volumetric displays, the viewing volume is swept out by a curved surface. The shapes of such displays can be spherical spirals [Langhans et al. 1998] or helical [Bahr et al. 1996].

Recently, rotating swept displays have been developed to use light fields to display different imagery to each viewing angle [Cossairt et al. 2007, Jones et al. 2007]. The benefit of such displays is that they can produce viewpoint dependent rendering effects, such as shading, and hidden surface removal. However, such displays are not truly volumetric, as the perceived location of a voxel might not be the actual location where it is emitted.

Rotating volumetric displays are one of the more promising technologies to date, with current implementations being available commercially (Figure 2-6). This is the technology we use in this thesis (Section 1.4).

2.3.3 Static Volumetric Displays

Static volumetric displays are defined as displays which generate light in true 3D space, without the need for mechanical motion [Ebert et al. 1999]. With this technique, emissive voxels are provided at a large number of locations in a static setup, without the use of oscillating or rotating mirrors or screens. This can be accomplished by exciting a plasma filled medium within a display volume to produce a glow at a single point.

A proof-of-concept of this displays was implemented in 1971, where two faint spots of light were generated inside a transparent crystal of erbium-doped calcium fluoride, with the use of filtered xenon lamps as excitation sources [Lewis et al. 1971].

Based on this early work, Downing et al. presented a three-color static volumetric display with improved material using high powered laser diodes [Downing et al. 1996]. The laser beams intersect inside a transparent volume, which consists of a rare earth-doped heavy metal fluoride glass. Red, green, and blue voxels can be illuminated by sequential two-step resonant absorption. Imagery can be produced in 3D by scanning the point of intersection of the lasers in various 3D locations of the volume. The demonstrated work was implemented in a prototype sugar-cube sized volume (Figure 2-7).

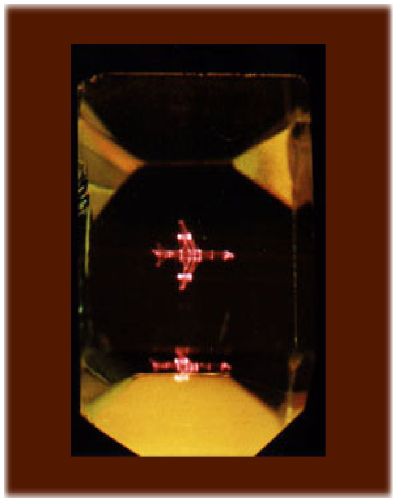

Figure 2-7. A prototype sugar-cube sized static volumetric display. Image taken from www.3dtl.com.

In another implementation, a gaseous volume was enclosed within a sealed glass container. Two diode lasers were intersected in Rubidium vapor [Kim et al. 1996]. The setup requires Rubidium to be heated within a vacuum, making it difficult to design for larger scale display volumes.

A more recent implementation used a pulsed laser to create glowing points of plasma in midair [Uchiyama et al. 2006]. This is one of the few volumetric displays which does not require a physical enclosure. However, the points of plasma are at an extreme temperature, so users are still unable to reach in and interact with the displayed imagery.

Although the current level of quality for static volumetric displays is not up to par with the current generation of rotating swept volumetric displays, their desirable properties make for a promising alternative which researchers continue to develop.

2.3.4 Summary

This section has outlined the most common technological implementations for volumetric displays. From an interaction standpoint, these technological differences may not seem so interesting; however, there are some interesting observations to be made. Most importantly, all of the discussed technologies share the properties which we have discussed in Section 1.2.

What is not common among the displays discussed in this section is the shape and size of the physical enclosure. For example, we have seen hemispherical domes, as well as cuboids. Thus, if the enclosure is to be used as an interactive surface, then the interaction techniques which are designed should be able to be implemented regardless of the shape of the enclosure surface. We will consider this in our designs presented later in this thesis, and also discuss how our results can be generalized to other display forms in Chapter 9.

2.4 Evaluation of Three-Dimensional Viewing Modes

In the previous two sections we have discussed the various display technologies for viewing 3D data. With all the available types of 3D display technologies, it is important for designers to know which will be the most appropriate device for their task at hand. Most relevant to our work, we will want to determine under what condition volumetric displays will prove to be beneficial. Researchers have tried to address this issue by conducting formal evaluations to provide comparisons of different viewing modes. We now provide a summary of such research, and an overview of the important results obtained to date.

2.4.1 Subjective Impressions of Three-Dimensional Viewing Modes

In an early study, Ware et al. compared the relative effectiveness of head coupled and stereoscopic views by allowing users to make comparisons between pairs of presentation methods [Ware et al. 1993]. Two different 3D scenes were presented to users. One consisted of a sphere with its shadow cast on a ground plane. The other consisted of a bent piece of tube based on Shepard and Metzler's mental rotation for 3D objects [Shepard and Metzler 1971]. Once a scene was presented, subjects could toggle between two presentation modes until they could decide which contributed more to the perception of the 3D space.

The experiment consisted of the following 5 conditions: perspective, stereo, head coupled monocular, head coupled binocular, head coupled with stereo. In the non-stereo conditions the same scene was presented to both eyes. In the binocular condition the viewpoint was between the eyes, while in the monocular condition, the viewpoint was correct for the right eye, and the subjects were asked to close their left eye. In the fixed viewpoint conditions, the perspective view was established by the initial head position at the start of the trial.

For each trial of the experiment, subjects were given one of the 10 pair wise comparisons of the 5 viewing conditions. Subjects toggled between the viewing conditions until they made their final selection, as to which was “best”.

Their results showed no differences between the two scenes so their data was combined. The results showed that users rarely preferred stereo without head coupling over head coupling without stereo. Another interesting result was that users picked head coupling without stereo over head coupling with stereo 68% of the time. The authors attribute this result to the ghosting effect in the stereo conditions, caused by imperfect phosphor decay, which causes cross talk between the left and right images.

Most importantly, their results supported the use of head coupled stereo viewing. All subjects said that they would use it for object visualization it were available.

2.4.2 Empirical Evaluation of Stereo and Motion Cues

The results of the subjective evaluations of three-dimensional viewing modes are unsurprising: providing stereo or motion cues will improve the user’s perception of a three-dimensional scene. Researchers have also tried to quantitatively determine exactly how much is gained by moving from 2D to 3D representations. Ware and Franck note that a completely general answer to this question cannot be expected, because the advantages will greatly depend on the specific task at hand [Ware and Franck 1996]. To gain a preliminary understanding of the issues, researchers have commonly chosen a path tracing task in a network of nodes, of which the results can be generalized to the large set of problems which can be represented in this way [Arthur et al. 1993, Sollenberger and Milgram 1991, Sollenberger and Milgram 1993, Ware et al. 1993, Ware and Franck 1996].

It has been shown that stereoscopy can improve user performance when detecting paths in a tree structure [Arthur et al. 1993, Sollenberger and Milgram 1991, Sollenberger and Milgram 1993, Ware and Franck 1996]. Sollenberger and Milgram [1991, 1993] also found that scene rotation further reduced errors in such a path tracing task. In their implementation, scene rotation was controlled by the system. In a follow up study, Ware et al. [1993] found similar results when the motion was controlled by the user, in the form of a head coupled perspective view.

In one of the more recent studies, Ware and Franck [1996] evaluated nine different types of viewing modes for a path tracing task. Randomly generated graphs were presented, and users were required to determine if two highlighted nodes were connected by a path of length two. The viewing modes consisted of different combinations of perspective, stereo, and motion parallax cues, where they tested both hand, head, a system controlled rotation of the scene.

It was found that the stereo viewing mode without motion was significantly worse than all three tested stereo modes with motion, including system controlled rotation, hand coupled rotation, and head coupled rotation. Of the three viewing modes which combined stereo and motion, there were no significant differences, showing that the motion parallax cues were important, but it did not matter how they were provided.

In the virtual reality literature, researchers have attempted to quantify the benefits of immersion within virtual environments. Barfield et al. found that head tracking significantly improves users' sense of presence [Barfield et al. 1997]. In a collaborative task, where two users had to maneuver a ring through a 3D object, Narayan et al. found that stereo was extremely important, while head tracking did not provide significant performance gains [Narayan et al. 2005].

2.4.3 Qualitative Evaluation of Volumetric Displays

Because volumetric displays are still in the developmental stages, little evaluation has been conducted to provide a quantitative comparison to other 3D viewing modes. However, qualitative benefits associated with their use have been reported. Balakrishnan et al. provide a good summary of these benefits, which we now discuss [Balakrishnan et al. 2001].

The fundamental difference is that volumetric displays generate images in true 3D space, so imagery can be viewed within the displays, just as one would view real physical objects. The human viewer can use their natural physiological mechanisms for depth perception, such as true motion parallax and stereopsis through eye convergence and accommodation, without the need for supplementary hardware. Furthermore, the convergence and accommodation cues are consistent, unlike in stereoscopic displays. As a result, the documented problem of asthenopia should not occur from viewing volumetric displays.

Another advantage of volumetric displays which Balakrishnan et al. discuss is that they provide a 360° viewing angle. As a result, these displays can be viewed from almost any angle, and can be viewed by multiple users simultaneously. This makes volumetric displays particularly suitable for collaborative applications.

2.4.4 Quantitative Evaluation of Volumetric Displays

To date, only a few studies have been reported, and none provide conclusive evidence as to how volumetric displays compare to other existing 3D display technologies for depth perception tasks.

Rosen et al. [2004] found that users could identify deformations in three-dimensional objects with more accuracy on a volumetric display than on a 2D display. However, given that the 2D display used did not provide any stereo or motion cues, this study does not provide much insight as to how the volumetric display compares to the other, more relevant, 3D displays that are available today.

A study of air traffic control tasks [Van Orden and Broyles 2000] found that a volumetric display was not superior to other displays except in a collision avoidance task. Unfortunately, the study provides insufficient detail of the procedure used or insight into the cause of the results. For example, the users' viewing height relative to the display – a factor that could have a major impact on the results – is not reported for any of the viewing modes. This could be one reason why the study somewhat surprisingly found that in a height judgment task a 2D side view resulted in the worst accuracy of all the displays studied, when one might expect such a view to give the best performance for such tasks. This study also indicates that stereo displays are not beneficial over 2D displays, which is surprising given the previous work that demonstrates the benefit of stereo displays.

In another evaluation of volumetric displays, participants navigated a simple maze using a joystick [Tyler et al. 2005]. However, the comparison was to a 2D perspective display, and there was no clear advantage in overall performance times. Further, the maze and task were 2D, providing little insight into the value of volumetric displays for understanding the 3D scenes that they were designed to display.

2.4.5 Summary

In this section, we have provided an overview of the evaluations which have been performed on the various 3D display technologies. We have seen that both motion parallax and stereoscopic cues can increase users' experiences, both quantitatively and qualitatively. We have also seen that a system which provides only motion parallax and no stereopsis can perform better than a display with only stereopsis and no motion parallax, while a display which provides both will generally provide the best results.

As for volumetric displays, we have presented some early work in their quantitative evaluation. However, our review of the literature revealed little conclusive data on the performance of volumetric displays in comparison to other 3D display techniques for 3D perception tasks. Further work in this area will be required to provide more data in this regard.

2.5 Three-Dimensional Input Devices

We have discussed the main 3D display technologies which are being used, and the evaluations which have been performed on them. In the remaining sections of this chapter, we will discuss issues related to interacting with these displays.

Three-dimensional displays do not just add a new dimension to visualization; they also present a new dimension for interaction. Although the mouse has strong benefits, which has made it such a successful device in the 2D realm (see [Balakrishnan et al. 1997] for a discussion of these properties), the input device only provides two degrees of freedom. There have been numerous attempts to map the 2D input of the mouse to 3D control, however researchers have also looked at input devices which provide three or more degree-of-freedom input for interacting with three-dimensional displays. In this section, we summarize some of these input devices.

2.5.1 Six Degree-of-Freedom Devices

Ware and Jessome explore the use of a free-space six degree-of-freedom device for a three- dimensional positioning task [Ware and Jessome 1988]. They called the device a "bat", as it is like a mouse which flies (Figure 2-8a). The tracking device provides six degrees of freedom by sensing its x, y, and z positions, as well as its roll, pitch, and yaw. There are a number of commercially available implementations of such six degree-of-freedom tracking devices (e.g. www.ascension-tech.com, www.polhemus.com) and they have been commonly used for interacting with virtual environments, as they can be used to directly manipulate the position and orientation of virtual objects [Bowman and Hodges 1997, Hollerbach and Jacobsen 1993, Mine 1995a, Poupyrev et al. 1996]. Frolich and Plate [2000] augmented a bat with 3 rods representing the principal axes, which could be pushed and pulled to specify constrained motion along the corresponding axis.

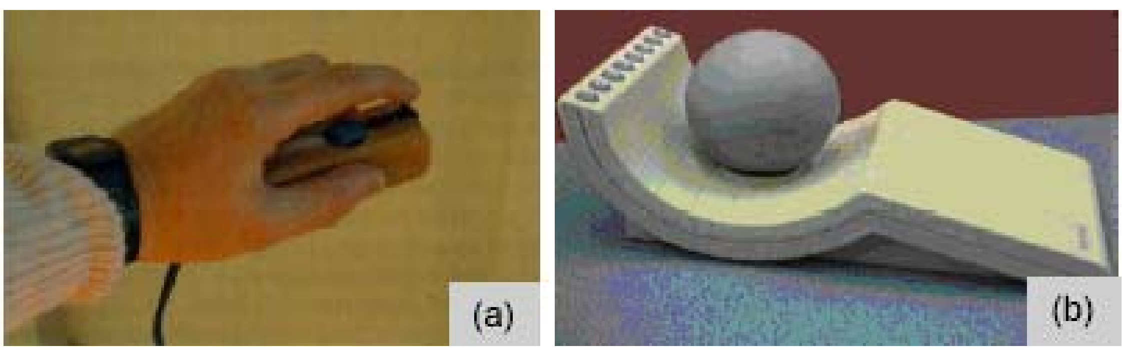

Figure 2-8. (a) The "bat" is an isotonic device consisting of a handle with a six degree-of-freedom tracker. (b) The spaceball ™ is an example of a six degree-of-freedom isometric device. Images taken from Zhai [1995].

Zhai gives a detailed comparison of these isotonic devices, which can be moved freely in 3D space, to six degree-of-freedom isometric input devices [Zhai 1995]. Isometric devices are mounted on a stationary surface, and sometimes termed "desktop devices" [Zhai 1998], and are like six degree-of-freedom joysticks. These devices are isometric as they all have some sort of spring loaded or elastic self-centering mechanism. Examples are the Spaceball™, SpaceMaster™ and SpaceMouse™ (Figure 2-8b). Zhai found that isotonic devices are much more appropriate for position control, while isometric devices are superior for rate control.

2.5.2 Roller mouse

A drawback of using the previously discussed six degree-of-freedom input devices is that they can be fatiguing to users. Unlike a mouse, a user cannot simply let go of the device, and have it maintain its position. To address this problem, 3D pointing devices have been developed, which are based on the standard 2D mouse, but enable 3D interactions. The "roller mouse" is a standard mouse with two wheels on the front, on either side of a single button [Venolia 1993]. The wheels are fixed to a common axle so that they control a single degree-of-freedom (Figure 2-9a). The roller mouse can be used to control a 3D cursor, by mapping the usual mouse movements to X and Y axis movements of the cursor, while mapping movement of the scroll-wheel to movement of the cursor along the Z axis.

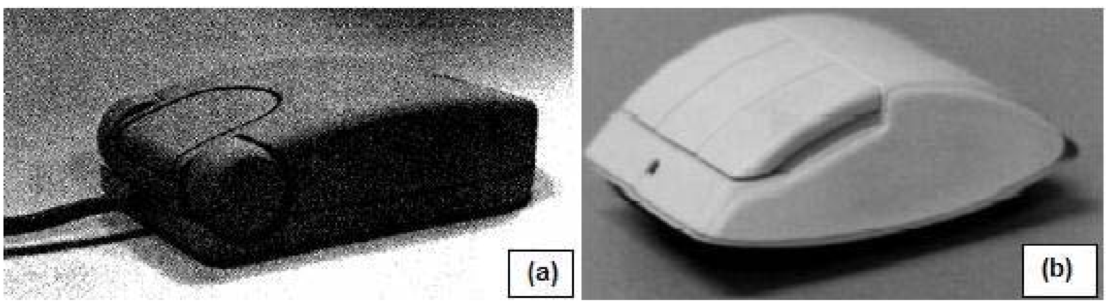

Figure 2-9. (a) The roller mouse is a standard mouse with two wheels for an added degree-of-freedom. Image taken from [Venolia 1993]. (b) The Rockin'Mouse has a rounded bottom adding two degrees of freedom. Image taken from [Balakrishnan et al. 1997].

2.5.3 Rockin'Mouse

The Rockin'Mouse is another example of a standard mouse which has been enhanced for added degrees of freedom [Balakrishnan et al. 1997]. The Rockin'Mouse can move like a standard mouse, but its bottom is rounded, so that it can be tilted left and right, as well as forward and backwards, providing two additional degrees of freedom (Figure 2-9b). In a 3D positioning task, it was shown that the Rockin'Mouse was 30% faster, when compared to a standard mouse.

Because the Rockin'Mouse and roller mouse operate the depth dimension by different behaviours and muscle groups from those which operate the X and Y dimensions, it may be difficult to produce simultaneous and coordinated motion, in comparison to six degree-of-freedom devices [Zhai 1998].

2.5.4 Bimanual Input

Researchers have also looked at using two devices, one in each hand, to increase the number of degrees of freedom available for 3D tasks. Zeleznik et al. explore a range of techniques for performing 3D operations in desktop applications, using two hands to control two independent cursors [Zeleznik et al. 1997]. Hinckley et al. embedded a head viewing prop and a cutting plane tool with six degree-of-freedom trackers, for use within a neurosurgical application [Hinckley et al. 1998]. The cutting plane tool could act on the head viewing prop, allowing for bimanual input. The 3-Draw system also used two props for designing 3D curves. A tablet and pen, used bimanually, were both tracked in 3D space [Sachs et al. 1991]. Balakrishnan and Kurtenbach performed two experiments to evaluate bimanual input for interacting with 3D scenes [Balakrishnan and Kurtenbach 1999]. They found that bimanual techniques were 20% faster for 3D selection, and strongly preferable in a free from 3D painting task.

2.5.5 High Degree-of-Freedom Input

By providing two devices, one for each hand, bimanual input techniques essentially multiply the number of degrees of freedom by two, allowing users to have more control of object parameters in 3D environments. Researchers have also looked at other methods for providing high degree-of-freedom input for 3D tasks which we now describe.

Krueger's VIDEOPLACE [Krueger 1991] was a deliberately informal and playful arena for the exploration of human interfaces with large displays. The user was in a darkened room with a real-time image of their silhouette displayed on a large video projection screen. Their entire body could be considered the input device, as the system responded to the user's image and motion with interactive graphics, video effects, and sound. While the interactions were with 2D data imagery on the large display, such fully body, or hand and multi-finger tracking could also be applied to 3D interactions [Cutler et al. 1997, Pierce et al. 1997, Schkolne et al. 2001].

Another interesting input device which has been used for 3D interfaces is called ShapeTape™. ShapeTape™ is a curve input device, which senses the bend and twist along its entire length. Researchers have explored the use of such a high degree-of-freedom input device for the manipulation of curves and surfaces in 3D environments [Balakrishnan et al. 1999, Grossman et al. 2003].

2.5.6 Summary

In this section we have provided an outline of the most common input devices used for interacting with 3D environments. Six degree-of-freedom devices can be used to provide direct manipulation of virtual objects, while enhancements to the mouse can also increase the number of degrees of freedom of an input device, while keeping some of the desirable properties of the mouse.

For added degrees of freedom, researchers have also looked at bimanual techniques, where the user can directly use both hands, or use both hands to hold physical tools. We have also seen how specialized high degree-of-freedom input devices can be used for both 2D and 3D interaction.

Because little research has been conducted on volumetric displays, no one has really explored what input devices should be used with these displays. It will be of interest to see which of these discussed input devices will be most appropriate for interacting with volumetric displays. We may find that the devices will need to be modified or enhanced. We may also find new methods of input, such as using the volumetric display's enclosing surface as a touch sensitive input device.

While the choice of input device is interesting, this will only be a step in the direction of developing interactive user interfaces for volumetric displays. In the following sections, we will see the 3D interaction techniques and applications which have been developed using the input devices discussed in this section.

2.6 Three-Dimensional Interaction Techniques

Along with finding appropriate input devices for volumetric displays, we will be interested in finding effective interaction techniques for such displays. In exploring this topic, the vast literature on interaction within 3D user interfaces and virtual reality environments is clearly relevant. A number of researchers have proposed and agreed upon four fundamental tasks for 3D interactions: navigation (or travel), selection, manipulation, and system control [Bowman and Hodges 1999, Mine 1995a, van Dam et al. 2000]. In this section we summarize the interactions techniques which have been developed in each of these categories.

2.6.1 Navigation

Navigation refers to the task of positioning and orienting the viewpoint within a 3D environment. Because the environment size displayed within a volumetric display is limited to the size of its enclosure, users will be able to effectively control their viewpoint orientation and position by simply changing the position of their head. As a result, the navigation techniques developed for vast large virtual environments, where changing the viewpoint position may require significant travel distances, will be less applicable, in comparison to the other three fundamental tasks, so we online give a brief outline of them here.

2.6.1.1 Cinematic Camera Metaphor

In non-immersive 3D environments, such as 3D graphics and animation programs, mouse movements coupled with modifier keys are generally used to manipulate the position of the virtual camera, which in turn controls the user's viewpoint of the scene. This is generally termed as a cinematic camera metaphor, since the user can tumble, track and dolly a viewpoint [Burtnyk et al. 2002]. The Unicam allows users to manipulate the position of the 3D camera using a mouse and only a single button, coupled with different gestures for invoking camera functionality [Zeleznik and Forsberg 1999]. The StyleCam allows authors to significantly tailor the best possible camera positions and paths to be used by a viewer [Burtnyk et al. 2002].

Various metaphors for viewpoint motion and control in 3D environments, which use higher degree-of-freedom input, have also been proposed. Ware and Osborne explore 3 such metaphors which were implemented using a six degree-of-freedom input device which we now discuss [Ware and Osborne 1990].

2.6.1.2 Eyeball-in-Hand Metaphor

With the eyeball-in-hand technique, the position of the input device is directly mapped to the position of the virtual camera, or eyeball, within the 3D scene. The viewpoint of this eyeball is then projected on the viewing display. They explored both an absolute mapping between the input device and eyeball position, and a relative mapping, in which the viewpoint only changed when a button was depressed. While the relative mapping allowed for ratcheting, it destroyed the user's mental model of a fixed invisible scene, and so users found it confusing. The chameleon discussed in Section 2.2.6 is an example of a display device which provides eyeball in hand navigation [Fitzmaurice 1993].

2.6.1.3 Scene-in-Hand Metaphor

With the scene-in-hand metaphor, the three-dimensional world is moved in correspondence with the six degree-of-freedom input device. If the device is rotated clockwise, then the scene will also rotate clockwise. Large movements can be accomplished by ratcheting, using a button as a clutch. It was found that the scene-in-hand metaphor was useful for changing the viewpoint in a hierarchal scene. Stoakley et al. designed an extension of the scene-in-hand metaphor, which makes use of a "World-in-Miniature" representation of the scene [Stoakley et al. 1995]. With this technique, navigation can be accomplished by manipulating a hand-held miniature copy of the virtual environment.

2.6.1.4 Flying Vehicle Metaphor

The other technique proposed is the flying vehicle metaphor. With this technique the scene is perceived from the position and orientation of the vehicle. The input device is used to control the spatial and angular velocity of the vehicle, allowing it to be moved throughout the scene. This metaphor is commonly used in 3D games, in which the "player position" can be thought of as the vehicle. The user controls the position and orientation of the viewpoint to move throughout the environment. This technique has been enhanced in various ways, such as allowing for rapid and controlled movement [Mackinlay et al. 1990].

2.6.2 Selection

Selection is a fundamental aspect of any interactive 3D system, as a user must be able to specify an object in the virtual world, so they can then manipulate or interact with it. Techniques for selection have been thoroughly developed within the VR literature. These techniques will be of high relevance to the design of interactive applications for volumetric displays. We now give a review of selection techniques used in 3D environments.

2.6.2.1 Image Plane Selection

When 3D environments are displayed on desktop screens, the monitor defines a 2D image plane, which displays the 3D scene. The application projects the objects in the 3D environment onto the image plane for display to the user. The user can select objects in the 3D scene using the image plane by positioning the mouse cursor over the object's projection on the image plane. In essence the cursor defines a ray perpendicular to the surface of the image plane, and clicking selects the first object which the ray intersects [Pierce et al. 1997]. This technique does not directly extend to 3D displays, as there is not always a well defined image plane. Ware and Lowther extend the technique for a stereoscopic display by defining the image plane as the viewpoint from only a single eye [Ware and Lowther 1997]. This "one-eyed cursor" was found to be an effective selection technique for 3D environments.

2.6.2.2 Ray Cursor

For selection within immersive 3D environments, Liang and Green [1994] implemented a ray firing selection mechanism that they called "laser gun" selection. A ray is emitted from the user's hand, so the user has control over the start point and orientation of the ray, much like a physical laser pointer (Figure 2-10a). The first object it intersected with would be selected. This technique allowed users to select both near and distant objects; however, they found it was difficult to select small and distant objects due to the angular accuracy required for such targets. To alleviate this problem, they created a mechanism called "spotlight selection". Instead of emitting a ray, the user emits a cone with its apex at the user's hand (Figure 2-10b). This allows a certain level of error to be accommodated.

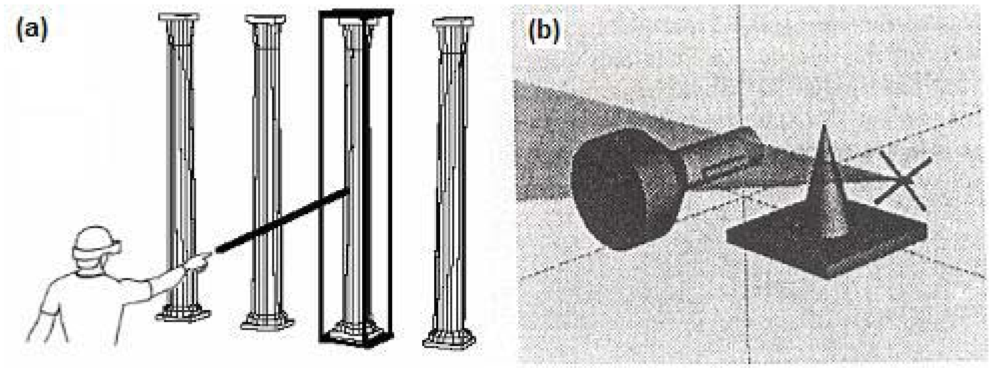

Figure 2-10. (a) Ray cursor selection. Image taken from Mine [1995a]. (b) Spotlight selection. Image taken from Liang and Green [1994].

A problem with the ray casting selection is that the ray can intersect multiple objects. This problem is magnified when a cone is used instead of a ray. Liang and Green developed a metric to decide which object would be selected if multiple targets were within the spotlight, based on the distance from the targets to the apex and central axis of the cone [Liang and Green 1994]. Hinckley suggests that the ray casting technique could be augmented with a mechanism for cycling through the set of all ray-object intersection points [Hinckley et al. 1994].

2.6.2.3 Occlusion Selection

Occlusion selection is an extension of the image plane selection used on a desktop to immersive 3D environments [Mine et al. 1997]. With this technique, users select objects that visually lie behind a hand-attached cursor. Essentially this is a ray casting technique, where the ray is defined by the vector which joins the user's eye and hand. This technique has been explored in a couple of different implementations.

Forsberg created aperture based selection where a cone is cast from the users eye and goes through the users finger, and the distance between the finger and the eye controls the angle of diversion of the cone [Forsberg et al. 1996]. Because orientation information is not used to define the direction of the ray, they used it for disambiguation when the ray intersected multiple objects. When the ray intersected more than one object, the object whose orientation was most closely matched by the input device was selected.

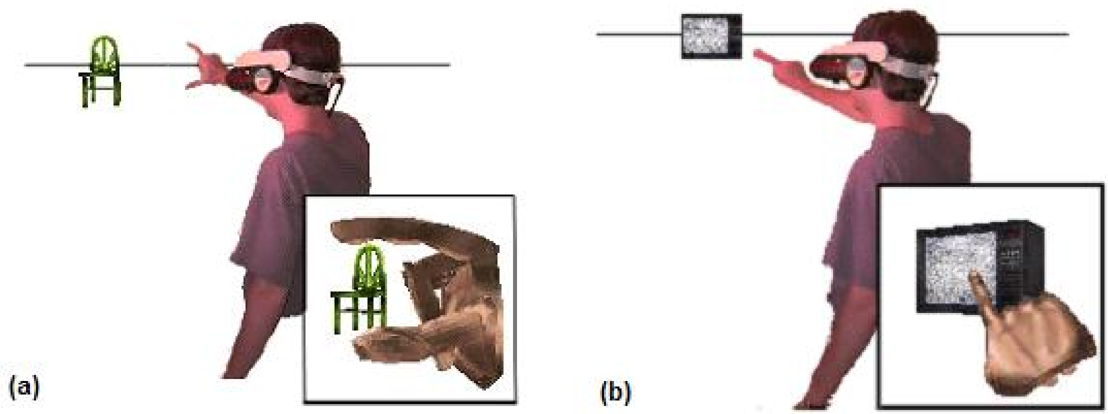

Figure 2-11. (a) The Head Crusher technique. Objects which lie between the user's finger and thumb can be selected. (b) The Sticky Finger technique. Objects occluded by the tip of the index finger can be selected. Images taken from Pierce et al. [1997].

Pierce et al. explored several different image plane interaction techniques which uses occlusion selection [Pierce et al. 1997]. With the Head Crusher technique, objects which lie between the user's index finger and thumb are selected (Figure 2-11a). The Sticky Finger technique selects objects by occluding them with the tip of the index finger (Figure 2-11b). The Lifting Palm technique allows users to select objects by flattening their hand and positioning their palm so that it appears to be underneath the object. The last technique was Framing Hands, in which both hands are used to define a bounding box around the object of interest, using the thumbs and index finger.

2.6.2.4 3D Cursor

A more direct method of interaction, using a 3D cursor which specifies X, Y and Z coordinates has also been suggested [Hinckley et al. 1994, Mine 1995a, Poupyrev et al. 1996]. Mine distinguishes between local and action-at-a-distance interactions in a virtual world [Mine 1995a]. He states that in local interactions, a direct mapping from the users hand to a 3D cursor should be used to select an object, while selection of objects at a distance, laser beam or spotlight selection could be used. He also suggests that a "virtual cursor or drone" could be moved by the user through the environment until it reaches a distance target for selection.

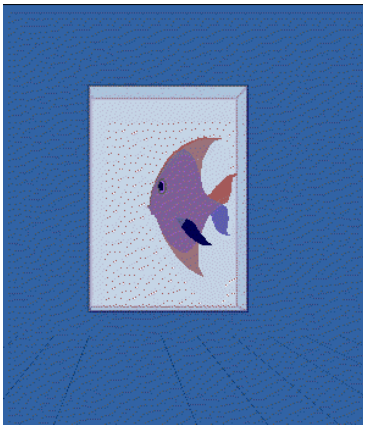

Figure 2-12. The silk cursor is a volume cursor with a semi-transparent lining. Image taken from [Zhai et al. 1994].

2.6.2.5 Silk Cursor

The Silk Cursor [Zhai et al. 1994] is a 3D volume cursor, the extension of 2D area cursors [Kabbash and Buxton 1995]. The silk cursor provides two advantages over a standard point cursor for selection tasks. Firstly, as with 2D area cursors, the effective activation area of a potential target is increased. Secondly, the rectangular shaped silk cursor is lined with a semi-transparent surface, creating a partial occlusion effect (Figure 2-12). This allows users to tell when a target is in front, inside, and behind the silk cursor. In a controlled experiment, the silk cursor was found to be beneficial in both monoscopic and stereoscopic displays.

2.6.2.6 Go – Go Technique

Poupyrev [1996] introduced the go-go selection technique to enhance 3D cursor selection for objects at a distance. The technique uses a metaphor of growing a user's arm interactively, with a non-linear mapping for selecting distant objects. When the user operates on nearby objects, the movement of the virtual hand matches movement of the real hand. To reach remote objects the user simply extends the hand further than 2/3 the length of their arm. The mapping becomes non-linear and the virtual arm grows. Unlike a standard 3D cursor, in which the range of selection is limited to arm's length, the go-go technique allows for selection of both nearby objects and those at a distance.

2.6.2.7 Evaluation of Selection Techniques

In a formal experiment, Ware and Lowther evaluated the one-eyed cursor, which is essentially a ray cursor, where the vector of the ray is always perpendicular to the surface of the screen [Ware and Lowther 1997]. They found it to be superior to a standard 3D cursor, as with the one eyed cursor, only two of the three dimensions of a goal target need to be defined. Bowman collected quantitative data to compare ray casting, occlusion selection and the go - go technique [Bowman et al. 1999]. The go-go technique was found to be significantly slower, as it required the positioning of the hand in 3D space. Although there was no significant difference in movement time between ray casting and occlusion selection, occlusion selection caused higher levels of arm strain. Ray casting was more comfortable as users could "shoot from the hip". To date there has not been a comparison of ray cursor with 3D volume cursors, such as the silk cursor, but area cursors have been shown to be advantageous in 2D environments [Worden et al. 1997].

While these results seem to indicate that ray casting techniques should be used for selection within volumetric displays, there are a couple of points which require further exploring. Firstly, the go-go technique was found to benefit from shortened target distances [Bowman et al. 1999]. Within a volumetric display, the distances will be limited by the physical enclosure, so 3D point techniques may not fare as poorly. Secondly, ray cursor selection breaks down in dense target environments, and to date there have not been any formal evaluations of techniques for disambiguating between multiple intersected targets. It may be the case that such a disambiguation mechanism will introduce an overhead cost which mitigates the advantages of ray cursors.

2.6.3 Manipulation

Once objects are selected, it is important that users can manipulate them in an interactive 3D application. We now discuss the techniques which have been developed to manipulate objects in 3D environments.

2.6.3.1 Direct Six Degree-of-Freedom Manipulation

One of the most common forms of manipulation in virtual environments is to map the movement of a six degree-of-freedom input device to the movement of a selected object

[Bowman and Hodges 1997, Hollerbach and Jacobsen 1993, Mine 1995a, Poupyrev et al. 1996]. This allows the object of interest to be rotated and translated as desired.

Techniques have also been proposed to allow for direct manipulation with objects which are at a distance. In the ISAAC system, users could first select an object with a ray cursor and then manipulate the object with a direct mapping from the hand [Mine 1995b]. A similar technique called HOMER (Hand-Centered Object Manipulation Extending Ray-Casting) was suggested by Bowman and Hodges, where the object is selected with a ray cursor and then a virtual hand jumps to the object position, and the object becomes attached to the hand [Bowman and Hodges 1997]. The World in Miniature interface allows users to quickly select and manipulate an object positioned anywhere in the environment by grabbing the miniature object which is in the palm of their hand [Stoakley et al. 1995]. They can orient the miniature environment by rotating the hand that holds it, allowing them to select objects that may be obscured from their current point of view. Once an object is selected, it can be positioned and oriented in the miniature world with a one-to-one mapping from the hand, or also at a greater scale for more fine grain control.

2.6.3.2 Voodoo Dolls

Voodoo dolls extend these direct manipulation techniques by allowing bimanual interactions [Pierce et al. 1999b]. With this technique, each hand can hold a voodoo doll, a miniature copy of an object in the virtual environment (Figure 2-13). The dolls are created by using occlusion selection with either hand, and held with a pinch. Dolls can be passed from one hand to the other, and dropped by releasing the pinch. The object held in the dominant hand can be positioned and oriented relative to the object being held in the non-dominant hand. This follows Guiard’s bimanual theory where the dominant hand works in the reference frame set by the non-dominant hand [Guiard 1987]. To move a lamp onto a table, for example, the user creates a voodoo doll of the lamp in the dominant hand and of the table in the non dominant hand. The user then places the miniature copy of the lamp on top of the table. In the actual virtual scene, the lamp will move on top of the table, which remains stationary. In a follow-up study, it was found that the Voodoo

Dolls technique allowed users to position and orient objects more precisely than the HOMER technique [Pierce and Pausch 2002].

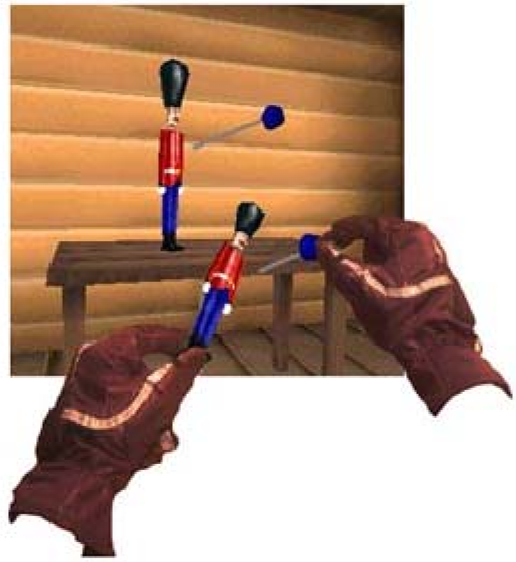

Figure 2-13. Voodoo dolls. The dominant hand manipulates a pin relative to the position of the toy soldier held by the non-dominant hand using hand-held miniature copies. Image taken from [Pierce et al. 1999b].

2.6.3.3 Three-Dimensional Widgets

Conner et al. explored manipulations of objects which diverged from the stranded direct manipulation techniques described above, using three-dimensional widgets [Conner et al. 1992]. Using the widgets allow for manipulations which are difficult or impossible to accomplish using a direct mapping from the physical hand, and also allow for constrained manipulations, such as rotating an object about a single axis (Figure 2-14). They describe several widgets which allow for different types of manipulations. For example a virtual sphere can be used to rotate an object. The mouse clicks and drags on the sphere, and the 2D mouse coordinates are mapped to points on the surface of the sphere, causing the object to rotate. Object handles are 3D widgets that constrain the manipulations. For example, handles can be available to scale, rotate, or translate an object along any of the major axes. Many current applications (e.g. MAYA™, 3D StudioMax™) for 3D modeling and animation make extensive use of such 3D widgets since they can be easily operated with status-quo mouse & keyboards input.

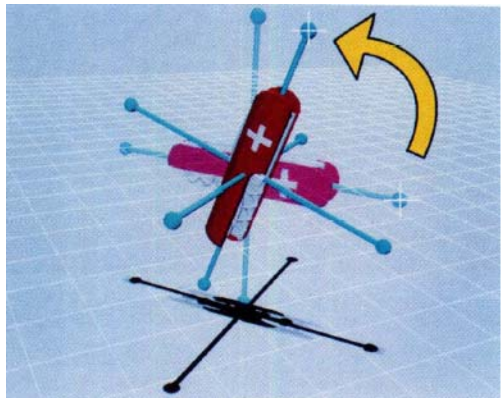

Figure 2-14. A three-dimensional widget. Object handles can be manipulated to perform a constrained rotation about a single axis. Image taken from Conner et al. [1992].

2.6.3.4 Gestural Manipulation

In Charade freehand gestures were used to manipulate 2-dimensional computerized objects in an augmented reality system [Baudel and Beaudouin-Lafon 1993]. Although the gestural interaction was mapped to the manipulation of two-dimensional objects, this could be a valuable method for manipulating three-dimensional objects. Since this foundational work, gestural interfaces have been developed for pen based devices [Zeleznik and Miller 2005], large displays [Vogel and Balakrishnan 2004], and tabletop displays [Wu and Balakrishnan 2003], and could be a valuable method for interaction with volumetric displays.

2.6.4 System Control

System control is the set of commands that the user gives to accomplish work within the application. For example, the user may wish to delete an object, or save a model which they are manipulating. In a standard desktop application, system control is achieved through toolbars and menus. Because these widgets are generally 2D, there are interesting challenges involved with integrating them into a 3D environment.

In Mine [1995b], a 2D menu was embedded in the virtual environment. The menu which was developed floats in 3D space and includes various widgets such as radio buttons, sliders, and dials. The user interacts with the menu using a ray cursor, so that the user does not have to make large reaching movements. In the JDCAD system [Liang and Green 1994] a ring menu was used for item selection, where the items were arranged along the circumference of a circle, and could be rotated until the item to be selected was directly in front of the user (Figure 2-15a).

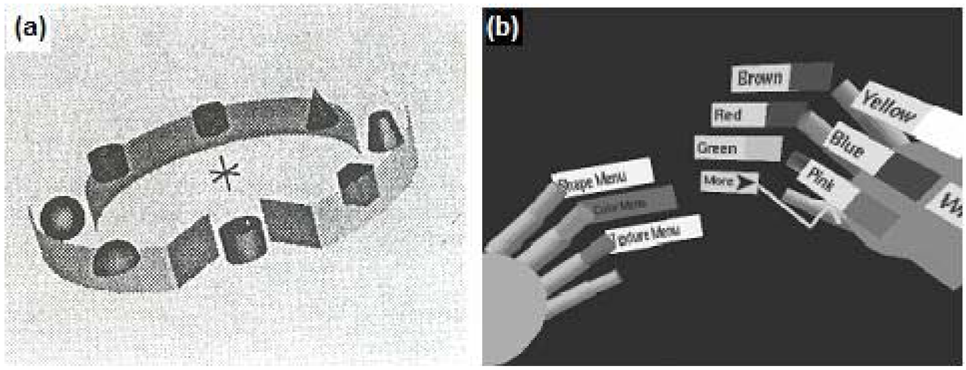

Figure 2-15. (a) Ring menus seen in JDCAD system. Menu can be rotated until item of interest is at front. Image taken from [Liang and Green 1994]. (b) The TULIP menu binds different menu options to each individual finger. Image taken from [Bowman and Wingrave 2001].

Bowman and Wingrave describe and compare three different menu techniques for virtual environments [Bowman and Wingrave 2001]. A floating menu was developed, in which the location of the menu is bound to the location of the user's head. The menu acts like a standard drop down menu, which can be navigated through using occlusions selection. The pen and tablet menu system places menus and widgets on the surface of a virtual tablet which corresponds to a physical surface which the user holds. A physical pen is held and controls a virtual stylus, which is used to interact with the tablet. TULIP menus require multiple-finger tracking, and binds different menu options to each finger. The user could select the options by pinching the thumb with the appropriate finger (Figure 2-15b). Of these three techniques, the pen and tablet menu was fastest, although users had a preference for the TULIP menu. Users also reported higher discomfort levels with the pen and tablet technique.

Figure 2-16. Volumetric display prototypes. (a) Selection using a laser pointer input device. (b) Cuboid shaped display allows user to extract side profile views. (c) 4-foot

large scale dome allows for collaborative use. Images taken from Balakrishnan et al. [2001]. All images are physical mock-ups and do not display virtual imagery.

2.6.5 Interaction Techniques for Volumetric Displays

Given that volumetric displays have not been easily available until recently, there has been relatively little research on how to use such displays effectively in an interactive manner. A speculative paper discusses possible interaction scenarios for volumetric displays, using wizard-of-oz mock-up prototypes to demonstrate various techniques for selection, displaying text and menus, and manipulating objects [Balakrishnan et al. 2001] (Figure 2-16). However, they did not have or make use of a real volumetric display and as such did not demonstrate any working implementations of their ideas. Some of the main conclusions of their research were as follows:

- *Physical rotation*: users were compelled to physically rotate the displays to get different views of the 3D content.
- *Touching the enclosure*: Users wanted to interact with internal objects by touching the surface of the displays as if it were a touch sensitive input device
- *Volume Management*: Given the realism of the displays, users wanted to manage the space inside with hand gestures which would be used to manage physical volumes
- *Viewpoint independent user interface widgets*: Since users are able to move around the displays, interface widgets should be available from any viewpoint.
- *Reuse*: Some input devices and interaction techniques from the VR literature transfer well to volumetric displays and should be reused. Other techniques require some enhancements and innovation so that they are appropriate for use on volumetric displays.

2.6.6 Summary

In this section we have presented a number of fundamental interaction techniques which have been developed for 3D environments. These interaction techniques are categorized as navigation, selection, manipulation, and system control. The navigation techniques will be of less relevance to volumetric displays, since there is a limited viewing volume, and users can easily change their viewpoint by physically moving. However, the development of selection, manipulation, and system control techniques will play a critical role in the development of user interfaces for volumetric displays.

As discussed in Section 2.6.5, a number of the techniques developed in the VR literature can be reused in volumetric displays. However, a fundamental difference from VR environments is that in volumetric displays, there is a physical barrier separating the user and the virtual data. As a result, many of the interaction techniques, such as direct grabbing and manipulation, will need to be modified, and new interactions techniques may be developed.

The motivation for the development of the techniques discussed in this section was to basically create an equivalent of the standard desktop GUI in a virtual environment. Our motivation will be similar, as we would like to develop the fundamental interaction techniques which could make up a fluid interface within a volumetric display. In Section 2.7, we will look at some of the interfaces which have been designed for 3D environments, which will help guide the development of applications for volumetric displays.

2.7 Interactive Three-Dimensional Applications

The interaction techniques discussed in the previous section will provide valuable insights for the development of interaction techniques for volumetric displays. Once new techniques are developed for the volumetric display, the next challenge will be to integrate these techniques into single fluid applications, which can be used for specialized purposes. We now provide a survey of some interactive 3D applications which have been developed for various 3D display systems, which will provide inspiration for the development of interactive applications for volumetric displays.

2.7.1 Designing Surfaces in 3D

One of the earliest interactive 3D applications was an experimental system for designing surfaces in 3D, built by Clark in the 1970's [Clark 1976]. It used a head-mounted display to provide a stereoscopic view, and the user interacted with the system using a three degree-of-freedom wand, which had its position tracked by three microfilament lines.

The wand could be used to drag control points in the 3D environment, and the system created curves and surfaces by interpolating through the control points.

2.7.2 3-Draw

Another system which is used to create 3D models is 3-Draw [Sachs et al. 1991]. The 3-Draw computer aided design tool is a tangible user interface for creating systems of 3D curves and editing them using deformations such as stretching, cutting, bending, and erasing. It is a two handed user interface. The non-dominant hand holds a thin rectangular plate with a six degree-of-freedom tracking device attached to it, and it is used to specify the viewing perspective. The dominant hand is used to point or draw 3D curves with a six degree-of-freedom stylus (Figure 2-17). A user can directly draw three-dimensional curves of a model as if the model were sitting on the plate that they are holding and positioning with their non-dominant hand. The user can also draw two dimensional curves to specify the shape of their curve and then specify its endpoints on an existing model. This allows an artist to use their sketching skills to accurately draw the shape of a curve, without worrying about its scale, placement, and orientation.

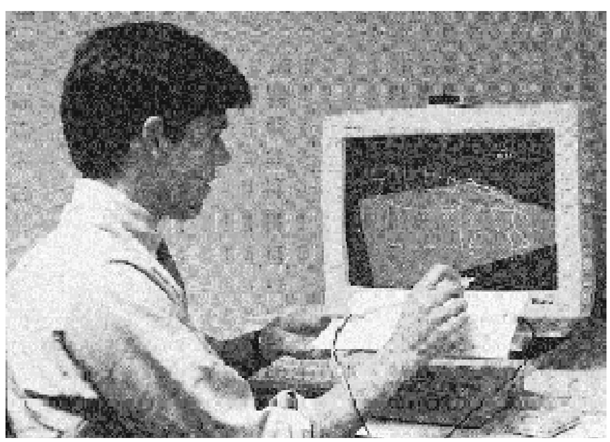

Figure 2-17. 3-Draw system. The dominant hand sketches 3D curves on the plane defined by the rectangular plate held in the non-dominant hand. Image taken from Sachs et al. [1991].

2.7.3 3DM

Butterworth developed a 3D modeling system called 3DM, which used a head mounted display to place the user inside the modeling space [Butterworth et al. 1992]. A six degree-of-freedom two-button mouse was used to interact with the system. Along with

3D models, 3DM provides interface elements such as a toolbox, which is a floating menu from which the main commands of the system can be accessed using a cursor. From the toolbox there are tools, which change the current mode of operation, commands, which perform single actions without changing the system's mode, and toggles, which change some global aspect of the system. The system provides various tools for surface creating, such as a triangle tool and an extrusion tool. The system also provides the standard editing operations, such as scaling, moving, cutting, and pasting. The user could navigate through the system by either walking or flying. The user could also change their scale relative to the environment to allow for faster navigation through larger distances.

2.7.4 JDCAD

Liang and Green developed a highly interactive 3D modeling system called JDCAD [Liang and Green 1994]. The system runs using a head tracked fish tank environment, and the user interacts with the system using a six degree-of-freedom bat. JDCAD provided a suite of general interaction techniques, and operations specific for 3D modeling. Users interface elements included a spherical pop-up menu called Daisy [Liang and Green 1993], and the spotlight selection cursor was used for object selection. Available modeling operations included 3D rubber banding object creation, reshaping, alignment, and clipping. It was found that when using the JDCAD system, it could take as little as 1/10 of the time required by a traditional system, to model a mechanical component of moderate complexity.

2.7.5 Interactive 3D Workbenches

Poston and Serra developed the Virtual Workbench [Poston and Serra 1994], which displays 3D images on a mirror in front of the user. The Virtual Workbench allows for bimanual interaction. Each hand can hold different props, which control various tools based on the current mode. The system was developed with medical applications in mind, including tools such as viewing the cross section of a volume (Figure 2-18a). The responsive workbench is another example of a two-handed interactive virtual reality system, in which the user wore a pair of pinch gloves [Cutler et al. 1997]. A stereoscopic image is displayed on a rear-projected tabletop display. A number of two-handed interactions were developed, such as two-handed zooming and rotation about an axis.

With the zooming technique, the non-dominant hand controls a focus point, while the dominant hand control the zoom factor. For rotation about an axis, the non-dominant hand defines an axis, while the dominant hand controls the rotation. A “steering wheel” rotation technique was also used, where both hands could grab the model and spin it (Figure 2-18b).

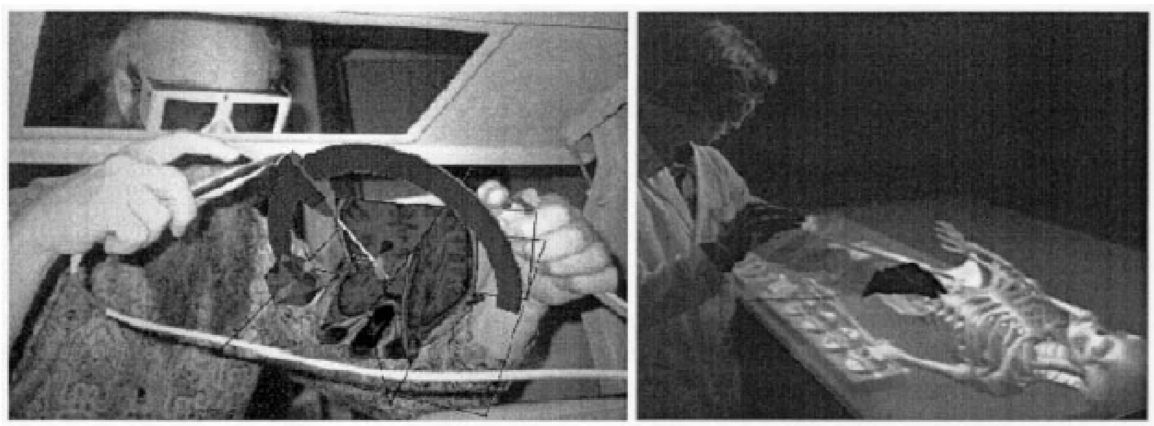

Figure 2-18. Interactive 3D workbenches. Left: the virtual workbench. Right: the Responsive Workbench. Images taken from Hinckley et al. [1998].

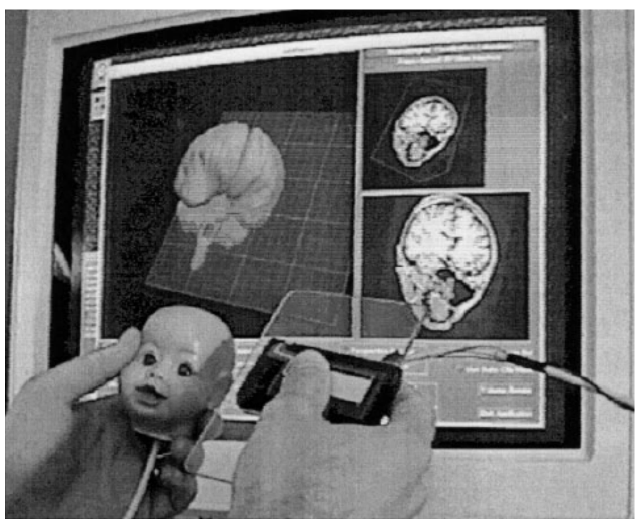

Figure 2-19. Two handed interface for neurosurgical visualization. Image taken from Hinckley et al. [1998].

2.7.6 Neurosurgical Planning

Hinckley et al. designed a user interface for three-dimensional neurosurgical visualization [Goble et al. 1995]. The interface was based on the manipulation of physical tools in free space. The interface was bimanual, with each hand being able to hold a tool (Figure 2-19). The interface allowed users to transfer their skills of manipulating tools with two hands to manipulating the interface props. A miniature head could be held in one hand, while the other hand could use a cutting plane or selection tool to slice open or point to certain locations of the head. A standard 2D monitor using a perspective projection was used for the system; however, it included a touch-screen, creating a 2D interface layer consisting of menus and widgets. The system was informally evaluated by neurosurgeons and it was found that the users could learn and understand the interface quickly.

2.7.7 Browsing Volumetric Data with Deformations

McGuffin et al. build upon Hinckley's work by supporting more complex browsing operations of volumetric data [McGuffin et al. 2003]. Instead of just "looking inside" volumetric data, such as seeing a specific cutting plane, users could perform various deformation operations to browse the data. Numerous deformation strategies were explored, such as peeling, slicing, and spreading parts of the volume . These operations could be performed on specific layers, allowing users to see particular areas of interest, while maintaining the context of the surrounding areas. The interaction techniques were controlled directly with pop-up menus and 3D widgets customized for each deformation technique.

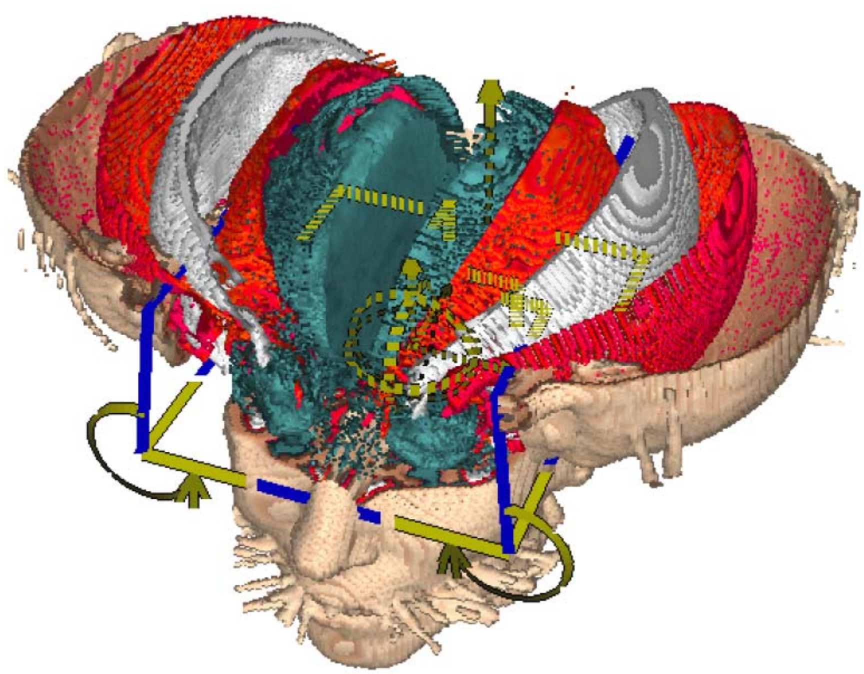

Figure 2-20. Deformations are used to browse volumetric data. Customized 3D widgets directly control the deformation technique. Image taken form McGuffin et al. [2003].

2.7.8 Summary

In this section we have presented a number of 3D applications which incorporate a variety of interaction techniques into single fluid interfaces. In each case, the tasks which the interfaces were designed for were appropriately matched with the system setup. For example, the interactive 3D workbenches discussed in Section 2.7.5 allowed for the exploration and manipulation of 3D medical data with both hands. Similarly, the neurosurgical application discussed in Section 2.7.6 allowed for bimanual interaction. Not only is the task of these applications appropriate, since these are cases in which it is important for users to understand the 3D structure of the data which they are working with, but the method of interaction was appropriate, as users of these systems, particularly doctors, would be used to working with both hands.

In the development of applications for volumetric displays, we will similarly seek out tasks which are particularly appropriate for the affordances of the display, and it will be important to create interaction techniques which will be appropriate for the intended users.

One such important affordance of volumetric displays is its 360° viewing angle, allowing multiple users to work with the system simultaneously. In the next section, we will discuss applications which have been designed for synchronous collaboration about a shared physical space, which will aid in the design of collaborative applications for volumetric displays.

2.8 Collaborative User Interfaces

One of the unique properties of the volumetric display is its 360° viewing angle. Not only does this allow a single user to view data from various locations, it also allows multiple users to view data from multiple locations, at the same time. This property could make volumetric displays particular promising for multi-user applications. The display itself, along with the data inside, would be a shared physical area which collaboration would be focused on. Early work on collaboration by a group on a shared display was performed by Nunamaker et al., with their development of decision systems to support group work [Nunamaker et al. 1991]. The specific scenario where collocated users interact with a *single* shared display was later categorized as single display groupware [Stewart et al. 1999]. This work also falls within the definition of some other categories of research. It fits Shafer and Bowman's definition of spatial collaboration - collaboration activities focused on physical areas [Schafer and Bowman 2005]. Following the taxonomy given by Ellis Gibbs and Rein, it would be classified as same time, same place collaboration [Ellis et al. 1991]. Baecker refers to applications supporting this form of collaboration as synchronous co-located groupware [Baecker 1994]. While collaborative user interfaces have not been explored on volumetric displays, there has been research relevant to the area, which we now review.

2.8.1 Single Display Groupware

Single display groupware (SDG) is generally defined as systems which support a group of collocated users working with a single display. Stewart et al. are credited with coining the term single display groupware [Stewart et al. 1999], but the first SDG system may have been MMM [Bier and Freeman 1991]. This system was the first to explore a system where multiple users shared a user interface not over a network, but across a single display. The MMM system looked at specific issues such as registering input devices with users, managing screen real-estate for the users, directing feedback to appropriate users without disrupting others, and allowing multiple users to work separately, without distracting each other. Following this work, Stewart et al. present a model for SDG, and discussed three issues central to SDG applications: Shared user interfaces, shared feedback, and coupled navigation [Stewart et al. 1999]. Shared user interfaces means that elements of the user interface must be accessible to each user, and user interface elements must be able to handle multiple simultaneous streams of input. Shared feedback refers to the fact that feedback provided to one user will generally be seen by all users, which can sometimes be problematic. Coupled navigation is an issue which arises when one user navigates to a different part of the viewed data or document. Generally all users' views are coupled and so other users will navigate as well. These central issues must be considered when designing a single display groupware application, including on the volumetric display.

2.8.2 Collaborative Tabletop Displays

Collaborative tabletop applications are good examples of SDG and spatial collaboration, as the focus of interaction is on a single shared physical device. The following is a small sample of research in this area.

2.8.2.1 RoomPlanner

Wu and Balakrishnan developed RoomPlanner, a prototype application for furniture layout [Wu and Balakrishnan 2003]. The application runs on a DiamondTouch display [Dietz and Leigh 2001] which can sense multiple input points from multiple users. They present a suite of interaction technique based on multi-finger and whole-hand gesture interaction. They also incorporate techniques specifically relevant for spatial collaboration, such as creating and maintaining personal interaction spaces, sharing information with collaborators, and interacting with private data (Figure 2-21).

Figure 2-21. RoomPlanner application. Users collaborate using multiple hand and finger gestures on the table top display. Figure taken from Wu and Balakrishnan [2003].

2.8.2.2 DiamondSpin

Shen et al. present a toolkit called DiamondSpin for experimentation with spatial collaboration on tabletop displays [Shen et al. 2004]. DiamondSpin allows for arbitrary orientation and positioning of documents on the display surface. The toolkit also supports various polygonal tabletop layouts, such as rectangular, octagonal, and circular. An important factor of the DiamondSpin is that it allows for multiple work areas, where multiple objects can be active concurrently, allowing for synchronous collaboration.

2.8.2.3 Storage Bins

Scott et al. developed a new interaction technique for tabletop displays called storage bins [Scott et al. 2005]. Storage bins are containers for data which allow users to share resources and transition between activities. Storage bins can be moved to bring a collection of items in and out of the users' current focus, and they can be expanded or collapsed to allow people to dynamically customize their working area. Storage bins provide the same capabilities as containers, allowing users to add or remove items as a group or individually. The bins can also be resized to vary the capacity of the bin.

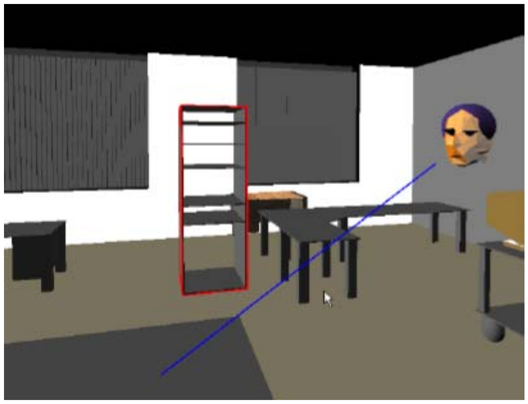

Figure 2-22. Users collaborate in a 3D environment. The avatar and blue line indicates the position and viewing direction of a collaborator. Image taken from [Schafer and Bowman [2005].

2.8.3 Spatial Collaboration in Virtual Environments

The relevant work on collaborative interaction discussed so far deals with interaction with 2D data – either in electronic meeting rooms or on digital table top displays. There has also been some relevant research on spatial collaboration in 3D virtual environments [Schafer and Bowman 2005]. Schafer and Bowman provide a virtual environment where users individually navigate through the 3D scene, while virtual avatars provide an indication of each user's location and orientation [Schafer and Bowman 2004]. They analyzed the effect of the frame of reference on the overall collaboration experience, and the users' awareness of each other's location and activities. They looked at an egocentric frame of reference, where each user has their own first person view, an exocentric view, where each user has an equivalent and shared third person view of the environment, and a combination of both, in which one user had an egocentric, and the other had an exocentric view. They found that when two users both had their own egocentric views, they could understand each other the best. This is a promising result for volumetric displays, as users would always have their own egocentric view, based on the position from which they were viewing the displays from. More recently, Schafer and Bowman [2005] developed a prototype system where users can arrange furniture in a virtual 3D environment. Users could switch between 2D and 3D views, and awareness features such as colored lines indicating other users' positions and viewing orientations are provided in both views (Figure 2-22). Their research showed that users took advantage of the multiple views, and the awareness features were essential for users to be able to understand what their collaborators were doing.

2.8.4 Text Layout and Orientation

One significant challenge with spatial collaboration, which is not addressed in the above research, is how text can be effectively displayed to multiple users. If users all have their own viewpoint of the shared physical area, then they will be viewing text from different angles. This could make text difficult, or sometimes impossible to read. For example, if two viewers of the volumetric display are standing 90° apart from each other, then 2D text facing one user will be completely parallel and thus invisible to the other user. The role of orientation has been studied in collaborative settings and it has been agreed upon that elegant algorithms for the orientation of screen elements are required to avoid depriving the users of rich interaction [Wigdor and Balakrishnan 2005].

On tabletop displays, there have been applications developed in which textual labels are oriented toward each participant [Agrawala et al. 1997]. Kruger et al. present a thorough review of systems which attempt to dynamically and automatically re-orient objects [Kruger et al. 2003]. Wigdor and Balakrishnan present a study of the effects of text orientation on table top displays, which can guide interface designers as to when it is important for text to be re-oriented [Wigdor and Balakrishnan 2005]. They found that the effect of orientation on reading labels, numeric and textual data, and performing serial searches, was less dramatic than what might have been previously assumed. This shows that in some situations the designer need not worry about providing perfect orientations of all text to every user of a system.

In the VR realm, generally each user has their own view of the world, and so there are not the same challenges for proving text orientations suitable for all users. The system can simply provide appropriate orientations for each user's view. Chen et al. provide a taxonomy of text layouts in virtual environments [Chen et al. 2003]. They note that text can either have a within-the-world display, in which it appears inside the 3D environment, and can even be projected on to the faces of various 3D objects, or it can have a heads-up display, in which the text appears on an invisible 2D image plane directly in front of the user's viewpoint. They found the heads-up display to perform better and to be preferable in a searching task. Bell et al. look at techniques for view management, in which decision algorithms for the layout of graphical objects such as text take into account visibility constraints that allow applications to manage what users see [Bell et al. 2001]. They provide an algorithm which continuously updates the position, size, and transparency of textual labels based on the users viewpoint, preventing objects from occluding each other. The technique they present is valuable for 3D environments which have a single viewer, but would need to be extended for when there are multiple viewers of the scene.

2.8.5 Summary

In this section we have provided an outline of the research which relates to the development of collaborative applications for volumetric displays. Such collaboration would be synchronous and co-located, and falls within the definitions of single display groupware, and spatial collaboration.

We presented a brief overview of the early single groupware research, and also looked at spatial collaborations occurring on table-top displays. Since users of such applications are focused on a shared physical area, albeit a 2D space, these interfaces will be of great relevance to the development of collaborative interfaces for volumetric displays. Some of the important interaction techniques for spatial collaboration discussed were the creation and manipulation of shared and personalized spaces, interaction with private data, supporting multiple active areas, and techniques for storing data and sharing data between users. Such techniques should be incorporated in collaborative applications for volumetric displays.

We also looked at relevant collaborations in VR environments. The research demonstrated that egocentric views can increase the understanding of what other users are doing. It was also seen that awareness features, such as illustrating the locations and areas of interaction of other users is important to enhance the sense of presence within such applications. It would be interesting to look at ways to do this on volumetric displays. Users would clearly know the location of the collaborators, but techniques for indicating what their collaborators were looking at or doing should be investigated.

Lastly we discussed the important challenge of orienting text and widgets appropriately for multiple users. We have discussed applications for table top displays which automatically orient data for multiple users. This is an especially important area for volumetric displays, as improperly oriented text could be difficult or even impossible to read, if parallel to a user's line of vision. Furthermore, users may not be stationary, taking advantage of the display's 360° viewing angle, and so menus and widgets should be accessible from all areas.

2.9 Summary of Background Literature

We have presented various theories, studies, and techniques from previous literature which relate to the exploration of user interfaces for volumetric displays. However, much of the research is taken from the virtual reality community, and there is very little previous research which has been focused on volumetric displays. An investigation into the interaction issues which are associated with volumetric displays will need to be conducted. We carry out such an investigation in the remainder of this thesis.

This investigation requires both a study of low-level human factor issues, as well as higher level explorations of interaction techniques and user interfaces. The most relevant low-level human factors issues are evaluating the viewing experience provided by the volumetric display in comparison to other 3D display modes, and investigating the user's ability to interact with a true 3D display space. These issues are investigated in Chapters 3-5. At a higher level, new interaction techniques and user interfaces will need to be explored, which are designed specifically for volumetric displays, for both single and multiple users. We describe our work in this area in Chapters 6-8.

3. Evaluation of Depth Perception

"We have deep depth."

-Yogi Berra

3.1 Introduction

[1]In our introduction we discussed a number of interesting and unique properties of volumetric displays, making them a potential platform for 3D applications. Maybe the most inherent property is the true 3D nature of the imagery which is displayed. A direct result of this property is that the depth cues which the user receives are consistent with one another. This is in contrast to traditional stereoscopic displays, where, most notably, there is a discrepancy between the convergence and accommodation cues. This raises an important human factors question: Are there any quantitative benefits to viewing imagery on the volumetric display, in comparison to traditional 3D stereoscopic displays?

There are numerous aspects of a display which can be evaluated to assess potential "benefits to viewing imagery". For example, displays may provide various levels of sense of presence, cognitive understanding of environments, subjective impressions, and shape recognition. In this chapter we focus on one of the lower level aspects of 3D displays: the perception of depth. There are three reasons why we specifically consider depth perception, and not some of the other, higher level, aspects. First, the main benefit which we would hope to obtain, by moving to a truly 3D display, is to have better perception of the 3D data. This could only be the case if the display increases the user's ability to

[1] A preliminary version of this work is published in Grossman and Balakrishnan [2006a].

perceive depth information. Second, in understanding viewing quality at the low level of depth perception, we can in turn hypothesize about some of the higher level properties. For example, it seems unlikely that there can be excellent recognition of shape without an improved perception of depth. The last reason is that the current technological state of volumetric displays prevents a higher level evaluation of its viewing properties. While the display quality is sufficient for performing low-level depth perception studies, higher level studies involving large 3D structures or environments may bias against the volumetric display only because of a poor image quality.

In this chapter, we outline a formal experiment comparing user performance in three different 3D depth perception tasks on a volumetric display, a standard perspective display, and a stereoscopic display with and without head tracking. We discuss the results of the study, which provides us with a better understanding of the perceptual benefits and drawbacks of volumetric displays. This discussion will be valuable in informing designers as to when a significant benefit will be gained by implementing their application on a volumetric display.

3.2 Related Work

3.2.1 3D Display Techniques and Depth Cues

One of the biggest challenges to creating realistic depictions of 3D scenes is to provide sufficient depth cues to enable the user to garner appropriate 3D information about the scene. In Section 2.2 we discussed a number of 3D displays which have been previously developed. These displays all provide users with a set of depth cues to improve the users' perception of the three-dimensional imagery. These depth cues are relevant to our investigation of depth perception in volumetric displays, since the quality of those cues will directly impact the user's ability to perceive depth. Here we summarize the main depth cues and display types which possess those cues.

One of the simplest and most effective depth cues is a perspective projection, which inversely scales the image coordinates with depth. Perspective is particularly effective when the scene has parallel lines [Brooks 1988]. Generally, perspective projections are provided from a single viewpoint, and users must vary this viewpoint in order to get a sufficiently rich understanding of the displayed 3D scene. Any desktop display which renders a 3D scene using a perspective projection can be considered a perspective display.

Stereopsis, resulting from binocular disparity, is another depth cue that can be exploited by presenting different images for each eye. A fused 3D image appears at the point of convergence between these two images. Generally, stereo displays require users to wear glasses, which separate the images for each eye. A review of such displays is given in Section 2.2.2.

Another important depth cue is motion parallax, obtained when the user's viewpoint of a 3D scene changes. Several systems have explored motion parallax cues [Diamond et al. 1982b, Fisher 1982, Sollenberger and Milgram 1991, Sollenberger and Milgram 1993], where the user's head position is tracked, and the location of the user's eyes is estimated by offsetting them by a constant distance from the user's head [Arthur et al. 1993]. The viewpoint of the virtual scene is then updated appropriately.

Systems which display 3D imagery can provide some or all of the above depth cues. For example, a fish-tank VR environment [Ware et al. 1993] is a head-coupled stereo display which provides all of these cues, with a viewing volume roughly equivalent to the inside of the monitor. We include a fish-tank VR display in our study as it is one of the more viable and least intrusive forms of 3D display currently available.

More immersive forms of 3D display technology also exist, usually in the form of head mounted displays [Buxton and Fitzmaurice 1998]. However, we will not include these forms of displays in the present study, since they have the drawback of separating the user from the real world [Arthur et al. 1993, Buxton and Fitzmaurice 1998]. While this may be acceptable for some tasks, we wish to compare the volumetric display to other displays which could be appropriately integrated into a regular workplace environment.

3.2.2 Evaluation of Display Techniques

In Section 2.4 we provide a thorough review of work which evaluates the relative value of different types of displays for 3D viewing. Here we summarize those studies which are most relevant to the work presented in this chapter, which specifically focus on depth perception.

It has been shown that stereoscopy can improve user performance when detecting paths in a tree structure [Arthur et al. 1993, Sollenberger and Milgram 1991, Sollenberger and Milgram 1993, Ware and Franck 1996]. Sollenberger and Milgram [1991, 1993] also found that scene rotation further reduced errors in such a path tracing task. In their implementation, scene rotation was controlled by the system. In a follow up study, Ware et al. [1993] found similar results when the motion was controlled by the user, with a head coupled perspective view.

McKenna [1992a] found head coupled perspective to also be beneficial in a 3D positioning task. Users were required to position a cursor to match the location of a cube in three dimensions. Results showed that a head coupled perspective gave the best results, while a mouse-controlled viewpoint condition decreased performance.

Arthur et al. [1993] showed that users' subjective impressions of a 3D scene were more positive when a head coupled without stereo display was used, over a stereo display without head coupling.

In a more recent study, Ware and Franck [Ware and Franck 1996] evaluated nine different types of viewing modes for a path tracing task. It was found that the stereo viewing mode without motion was significantly worse than all three tested stereo modes with motion, including system controlled rotation, hand coupled rotation, and head coupled rotation. Of the three viewing modes which combined stereo and motion, there were no significant differences, showing that the motion parallax cues were important, but it did not matter how they were provided.

While some initial studies have looked at evaluating the display quality of volumetric displays [Rosen et al. 2004, Tyler et al. 2005, Van Orden and Broyles 2000], results about their effect on depth perception are unclear. However, given that volumetric displays provide all of the above described depth cues, we would expect that depth perception on volumetric displays is at least comparable to the more traditional 3D displays. Furthermore, as discussed in Section 1.2.2, unlike the more traditional displays, volumetric displays provide users with consistent convergence and accommodation cues, which may lead us to believe that volumetric displays could improve users' depth perception. We use the following experiment to investigate this issue.

3.3 Experiment

3.3.1 Goals

The purpose of our study is to compare volumetric displays to other 3D display techniques with regard to user ability to perceive depth information when viewing 3D imagery. Our intention is not to run an exhaustive comparison of all 3D displays, but rather to provide data for volumetric displays, which is our display of interest, in comparison to established ones. Empirical data as to the capabilities of volumetric displays will be useful in guiding future user interface designs as the technology continues to improve.

Two factors are particularly important in designing this study: the type of display techniques to compare the volumetric display against, and the tasks used for the comparison.

3.3.2 Manipulation of Display Techniques

As outlined in Section 2.2, the number of different 3D display techniques that can be used for comparison is quite numerous. In a study by Ware and Franck [1996], for example, nine different display techniques were evaluated. To keep the size of our study manageable, we chose three display techniques to compare to the volumetric display, which were earlier found to differ significantly from each other in the Ware and Franck study. The first is a perspective projection on a 2D display, the second is a static stereoscopic display, and the third, is a stereoscopic display with head tracking. The third technique is closest to viewing the volumetric display, as both stereo and motion cues are provided. The literature shows that stereo with head tracking outperforms stereo only, which in turn outperforms perspective only [Ware and Franck 1996].

Including these three display techniques will allow us to determine where the volumetric display lies in this continuum of baselines. In the event that the volumetric display does not outperform the leading candidate – stereo with head tracking – we will be able to determine if the volumetric display is better than *any* of these three baselines.

We omitted hand coupled techniques from the study as previous research did not find them to result in significantly different performance from head coupled views. In all cases user input was through a keyboard, positioned in front of the display.

The display parameters were set such that apparent sizes were controlled to be equal across all displays. Other factors such as color, resolution, and brightness were set at optimal viewing values for each individual display. This allowed us to determine how the volumetric display compares to the baselines in a "best case" scenario for all displays. We now discuss each hardware setup.

3.3.2.1 Perspective Projection on 2D Display

A 19-inch Dell Trinitron CRT monitor was used, with the 3D scenes projected onto the 2D plane using a perspective projection. We used a perspective projection that simulated the viewing volume of the volumetric display, with the camera positioned at the actual user's viewpoint (approximately 20 inches away from the scene). The refresh rate of the monitor was 120Hz.

3.3.2.2 Stereoscopic Display

The same monitor as in the 2D perspective setup was used. A StereoGraphics CrystalEyes 3D LCD shutter glasses provided stereo viewing. Left and right eye images were provided by the top and bottom half of the frame buffer respectively. The monitor ran at 120Hz with each eye receiving a 60Hz update rate, coordinated with the shutter glasses. The glasses were synchronized with the display by an infrared transmitter positioned on top of the monitor.

3.3.2.3 Stereoscopic Display with Head Tracking

This setup was the same as the stereoscopic display, except that, in addition, the user's head position was tracked in real-time and used to generate the correct perspective view for each eye position. Head position was measured using an Ascension Flock-of-Birds electromagnetic six degree-of-freedom tracker. The tracker had a positional accuracy of less than 2mm, and operated at 100Hz. The tracker's electromagnetic receiver was positioned on the user's forehead using a headband. The position of each eye was derived from this reported head position, and the perspective view was continuously updated accordingly. Figure 3-1 illustrates this setup.

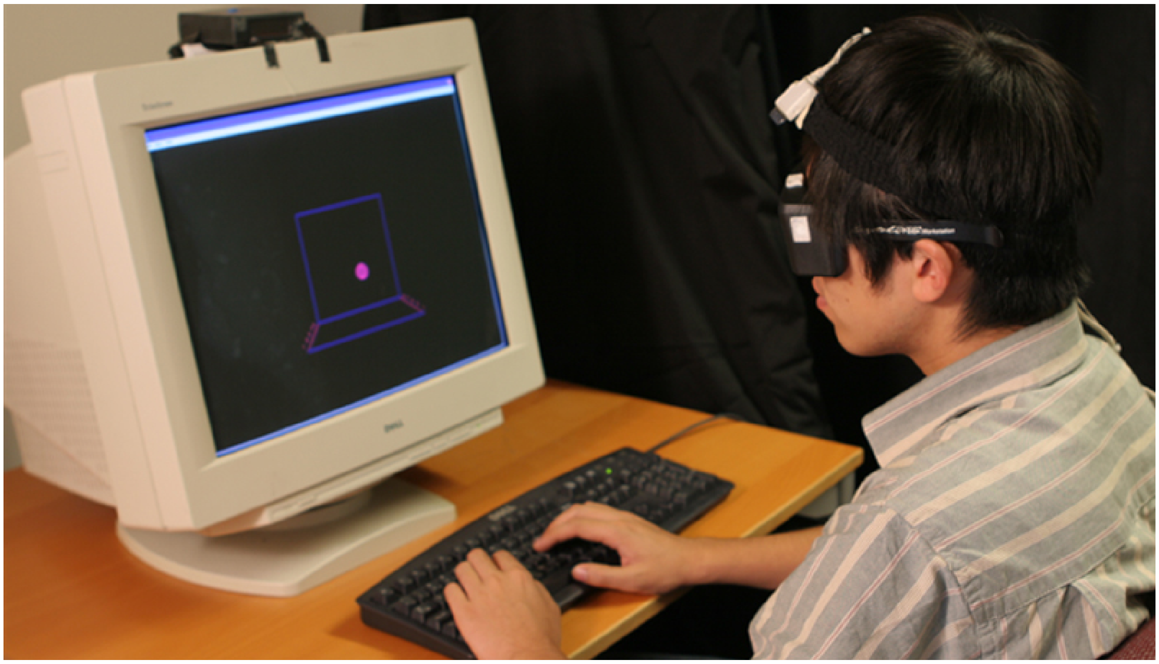

Figure 3-1. Stereoscopic display with head tracking setup. Shutter glasses provide stereo viewing, and an electromagnetic tracker provides head tracking.

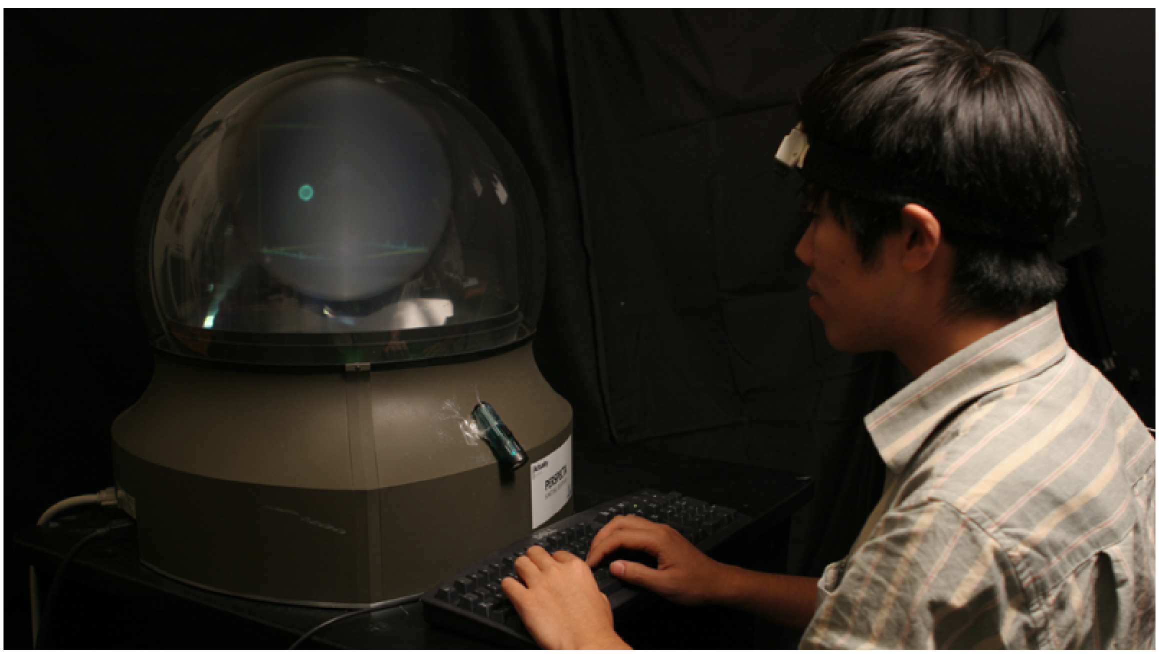

Figure 3-2. Volumetric display setup. The electromagnetic tracker measures head movement to allow imagery to be displayed only when head position is within a range.

3.3.2.4 Volumetric Display

We used a volumetric display from Actuality Systems (www.actuality-systems.com), which is described in detail in Section 1.4. It generates a 10” diameter spherical 3D volumetric image by sweeping a semi-transparent 2D image plane around the Y-axis. Each slice consists of 768x768 pixels, and a total of 198 2D images (slices) are uniformly displayed around the Y-axis, resulting in a total of 116 million voxels. The display's refresh rate is 24Hz. The user sat in front of the display, in the same chair used for the other three display setups (Figure 3-2). To ensure the same viewing angles were used for all displays, a calibration program was used to ensure that the user's viewpoint was aligned with the center of the display. In this and the previous stereo with head tracking technique, we limited the amount of head movement by fading out the 3D scene if the head position deviated from a calibrated starting position by more than 10 inches. Limiting the head movements prevented users from taking on viewpoints which would trivialize the depth perception tasks. Other than fading out the scene, head movements had no effect on what was being drawn in the volumetric display. Tracking head movements in the volumetric setup also allowed us to compare the amount of head movement used with what occurred in the stereo head tracking setup. Room lights were off to enable optimal viewing.

3.3.3 Tasks

Instead of using a single task to evaluate depth perception within a 3D scene, we chose a set of three tasks, all of which have been previously used to study human depth perception. Doing so allowed us to determine how the volumetric display compared to the other display techniques for a variety of task scenarios. The first task required participants to judge the depth of a single object in a 3D scene. The second required them to view and comprehend a complex graph which could be used for information visualization. In the third task, participants studied the position and heading of two moving objects and judged if they were on a collision course or not. In each of these tasks, the error rate was the main dependent variable, as previous studies have shown this to be highly sensitive to display technique [Arthur et al. 1993, Sollenberger and Milgram 1991, Ware and Franck 1996].

Participants were not required to perform any virtual object manipulations to complete any of the tasks, ensuring that the task was purely perceptual in nature. In a pilot study, we included a six degree-of-freedom docking task that had both perceptual and manipulation components. We found that the manipulation component added an overhead to task learning, completion times, and accuracy, while providing little additional insight into our primary goal of evaluating depth perception.

In another pilot study, we allowed participants to take as much time as they wanted to complete a task. However, some would take their time, while others seemed to rush through the experiment. Participants also took different amounts of time depending on the display technique. As an alternative, in all of the tasks, scenes were viewed for a controlled period of time, and we only measured accuracy. The period of time was chosen for each task to give the users a reasonable chance to provide an accurate response. These times were chosen based on our pilot study data. This design also prevented participants from "racing through the experiment", and, as such, they had no reason not to provide their best answers. We now discuss in detail the three tasks and their procedures and designs used during the experiment.

3.3.3.1 Task 1: Depth rating

In this task, participants were required to rank the depth of a sphere which was floating above the floor in a 3D graphical scene. This is similar to tasks previously used to evaluate the effect of shadows on perception of depth and spatial relationships [Hubona et al. 1999, Wanger 1992, Wanger et al. 1992, Yonas 1979]. In our implementation, square outlines of the floor and back wall of the scene were drawn. Along both sides of the floor, nine uniformly distributed tick marks were drawn, the first just after the beginning of the floor, and the last just before the back wall. A wireframe sphere was drawn floating above the floor. The center of the sphere was exactly aligned with one of the nine tick-marks. The task of the participant was to determine the depth of the sphere, by indicating which tick mark they thought it was aligned with. The numbers 1, 3, 5, 7, and 9 were drawn beside the corresponding tick-marks, but participants were told that 2, 4, 6, 8 were also valid answers. The even numbers were omitted from the display to reduce visual clutter. The horizontal and vertical viewing angles of the scene were both 0°, parallel with the floor of the scene, and the participant's viewpoint was centered with the middle of the back wall. This ensured that participants would have to rely on their depth perception to determine their answer. Figure 3-3 illustrates this task. For each trial, the sphere could take on any of the nine depth values, and the size, height, and lateral position of the sphere were systematically varied during the experiment.

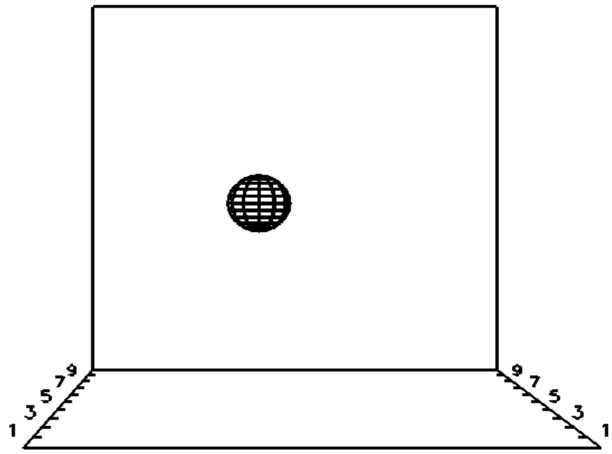

Figure 3-3. Depth rating task. Participant specified the depth (1-9) of the sphere's location in 3D space.

The procedure for this task was as follows: participants hit the space bar to begin a trial. The scene was then drawn, and a timer started. Participants had exactly 3 seconds to view the scene. We determined through pilot studies that 3 seconds was a reasonable duration for participants to comprehend the scene and make an assessment of the object's location in depth. After 3 seconds, the scene disappeared, and participants were prompted to enter their answer by hitting the relevant numeric key on the keyboard. To control viewing time, participants could not enter their answer until the scene disappeared.

Participants completed 54 trials, with the sphere appearing at each depth value exactly 6 times. The height, lateral position, and size of the sphere took on random values which were generated before the experiments began, so that each participant saw the same 54 trials, but in a random order. Before starting, four practice trials were given, so that the participants were familiar with the task.

3.3.3.2 Task 2: Path Tracing

The purpose of this task was to evaluate the effectiveness of the various displays for understanding a complex 3D information network represented by a graph. The ability to trace paths in such graphs is an important aspect to understanding such information networks. Such path tracing tasks have previously been used to evaluate various 3D viewing modes, such as perspective, stereoscopic, and head coupled stereoscopic [Arthur et al. 1993, Ware and Franck 1996] Thus, our use in this context is consistent with the literature. In particular, our implementation was similar to that used in Ware et al. [Ware and Franck 1996]. The computer generated a random 3D graph consisting of 36 nodes and 48 edges. The nodes were randomly placed in a 5" radius spherical volume. The nodes were divided into three groups of 12. Two of these groups were considered leaf nodes, while the third group was considered to be intermediate nodes. Each leaf node was connected to exactly two randomly chosen intermediate nodes, resulting in the 48 edges. In each graph, two leaf nodes were highlighted by in the use of a different color than the other nodes. The task of the user was to determine whether or not there was a path of length two from one highlighted node to the other. Because the highlighted nodes were both leaf nodes, they could never be connected directly. Figure 3-4 illustrates this task.

With this task, the participant had 8 seconds to view the graph before the scene disappeared. The 8 second parameter was determined via pilot studies as being a reasonable duration for comprehending the graph. Once the scene disappeared the participant would either hit 'y' or 'n' on the keyboard, 'y' for "yes there is a path of length two", or 'n' for "no there isn't a path of length two". The highlighted nodes were drawn before the participant hit the spacebar, so that there would be no difficulty finding them once the entire graph was displayed.

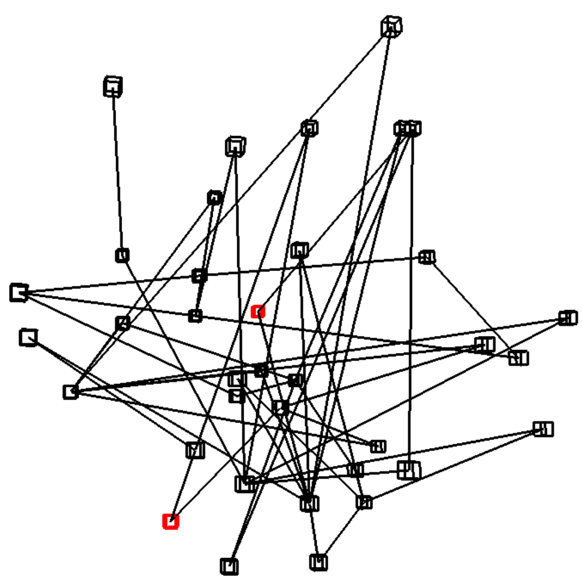

Figure 3-4. Path tracing task. Participant determined if there was a path of length two connecting the two highlighted nodes.

Twelve graphs with predetermined highlighted nodes were randomly generated before the experiments began. In six of these graphs, a path of length two existed, and in the other six, a path did not exist. Each graph was presented at 3 different orientations, rotated by 120° along the up-down axis, resulting in 36 trials. Each participant saw the same 12 graphs, and the order of the 36 trials was randomized between participants. Four practice trials were given: in two of these practice trials, a path existed; in the other two, a path did not exist. This ensured that participants thoroughly understood the task.

The volumetric display we used had certain regions where data was slightly harder to perceive, due to the physical boundaries of the spinning screen within the display. This could potentially make it hard to see important nodes and edges of a graph, depending on its orientation. By presenting each graph in three orientations, our intention was to mitigate potential perception difficulties due to the vagaries of our display. It is important to note that this is merely a safeguard against a weakness in the particular display we used, rather than a concern with volumetric displays in general. A detailed discussion of our display's viewing artifacts is presented later.

3.3.3.3 Task 3: Potential Collision Judgment

In this task, participants were asked to make spatial judgments about a *dynamic* 3D scene. We chose a task that had previously been used to evaluate display techniques and visual enhancements for 3D scenes [DeLucia 1995, Pfautz 2000, Van Orden and Broyles 2000]: two objects flew together and disappeared before reaching a potential point of impact, and participants determined whether or not the objects were going to collide (Figure 3-5a).

The objects were drawn as wireframe cubes, and the flight paths were always at the same altitude. The trajectory of each cube varied from trial to trial, but the depth component of the trajectory was always positive (moving away from the user), and the angle between the two 3D flight paths was always 90 degrees. The horizontal and depth coordinates of the two objects were varied to create three different scenarios. In one case, the cubes were on a true collision course (Figure 3-5b). In the other two cases, the cubes were on a near-collision course, such that one cube would eventually pass behind the other if the animation was allowed to continue (Figure 3-5c, d). The cubes disappeared before they collided or passed by each other when the horizontal distance between their positions in 3D space reached a threshold value of approximately 2.25cm. The size and speed of the cubes were always the same. The user's viewpoint position was equal to the altitude of the objects. This made the task particularly difficult, for if the animation continued, the objects would collide on the 2D projection of the scene, regardless of whether or not they were actually colliding in 3D space. For frame of reference, a ground plane grid was drawn below the objects flight path.

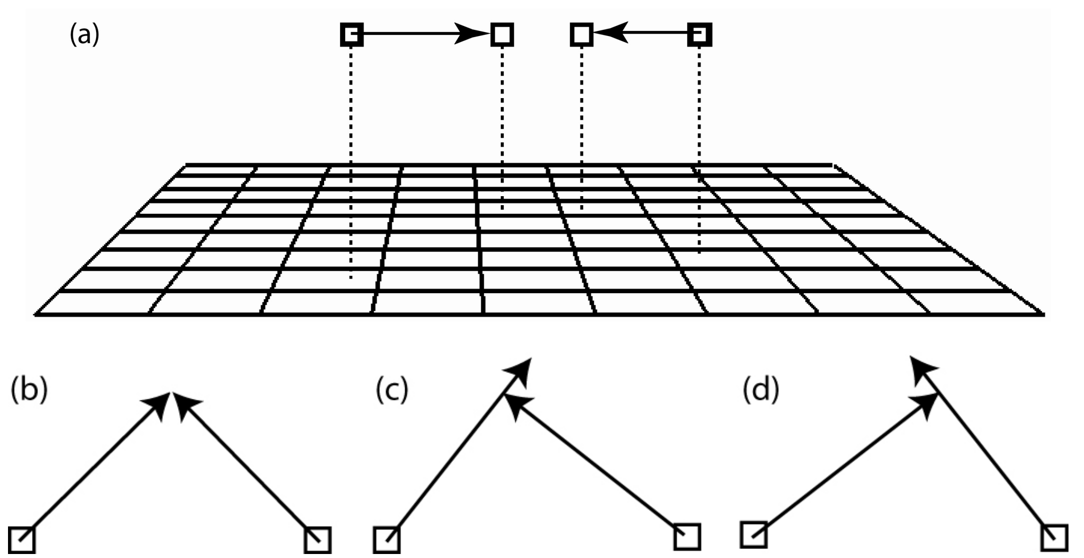

Figure 3-5 (a) User's perspective of the collision task (dashed lines and arrows are for illustration only). (b-d) Top view of the task, illustrating the three possible scenarios.

As in the previous two tasks, a trial began when the participant hit the space bar. The animation would then play, and when the horizontal distance between the two cubes reached the threshold value, the scene disappeared. At this point, users would either hit 'y' or 'n' on the keyboard, 'y' for "yes they were going to collide", or 'n' for "no they were not going to collide".

For this task, there were 60 trials. Trajectories were randomly generated before the experiment started, so that all participants saw the same 60 trials, but in random order. In 30 of the trials, the objects were on a true collision course. Trials, in which the left object would pass behind the right object, and vice- versa, appeared 15 times each. The 60 trials appeared in random order. Four warm-up trials were given. In two of the trials, the objects were on a true collision course. In the other two, each miss scenario occurred once.

3.3.4 Participants

Five female and seven male volunteers participated in the experiment. Participants were undergraduate students in computer science, and were screened for adequate stereo vision using the Stereo Optical RANDDOT stereopsis test. Participants ranged in ages from 18 to 25. None of the participants had previously viewed the volumetric display, and only one participant had significant experience with virtual reality systems. Most of the participants did have some experience with 3D video games, with 8 of the 12 participants reporting that they had lots of experience with such games.

3.3.5 Overall Experiment Design

The experiment was performed in one sitting, where participants completed all trials for one display technique before moving on to the next. The 12 participants were randomly split into 4 groups of 3. The presentation order of the four display techniques was counterbalanced using a Latin square design [Montgomery 2001]. For each display, participants completed the three tasks. The tasks always appeared in the same order, each lasting about 5-7 minutes. The entire experiment took approximately 75 minutes.

3.3.6 Results

3.3.6.1 Task 1: Depth rating

The main performance measure for this task was the error magnitude, defined as the average difference between the participant's answer and the correct expected answer. For our analysis we took the mean error rate at each object depth level, for each subject.

Analysis of variance showed that the display type had a significant effect on error magnitude ($F_{3,33} = 252.31$, $p<.0001$). The volumetric display had the lowest error magnitude of 0.48, which was a 25% improvement on the stereoscopic display with head tracking setup, which had the next lowest error magnitude of 0.64. Pair-wise means comparison using Tukey's post-hoc test showed this difference to be significant ($p<.05$). The perspective display was significantly worse than the other three techniques ($p<.0001$) while the difference between the stereoscopic display with and without head tracking was not significant (Figure 3-6).

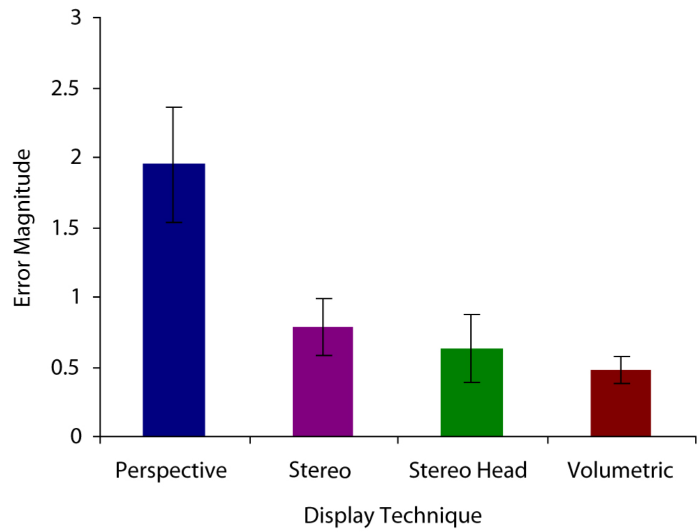

Figure 3-6. Error magnitudes for the depth rating task, with 95% confidence intervals.

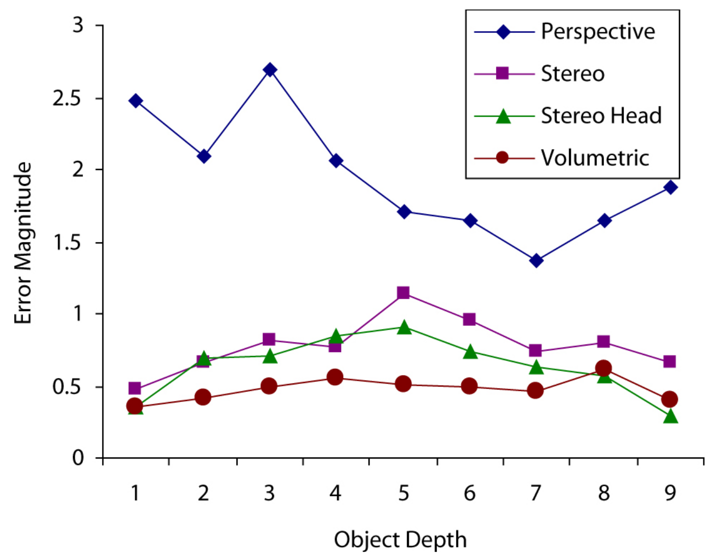

Figure 3-7. Error magnitudes for the depth rating task, for each object depth.

The depth of the object also had a significant effect ($F_{8,88} = 3.93$, $p<.0005$) on error magnitude (Figure 3-7). The two stereoscopic display techniques did best at the extreme values. Pair-wise means comparisons using Tukey's post-hoc test showed that for the stereoscopic display with head tracking, depth values 5 and 6 were significantly worse than depth values 1 and 9, and for the stereoscopic only display, depth value 1 was significantly different from depth value 5. This could be due to the fact that at the extreme values, the user could use the room (either the front of the floor, or the back wall), as a frame of reference, improving the results. Contrary to this, the results for the volumetric display seem to be fairly uniform throughout, with pair-wise comparisons showing no two values to be significantly different. The values for the perspective display are quite sporadic, likely due to users simply guessing the answer, as depth perception on these displays is quite poor.

3.3.6.2 Task 2: Path Tracing

The main performance measure for this task was the error rate, defined as the average number of errors per trial. Analysis of variance showed that the display type had a significant effect on average error rate grouped by participant ($F_{3,33} = 11.94$, $p<.0001$). In this task, the stereoscopic display with head tracking resulted in the best performance, with an average error rate of 12.7%. Error rates for the stereoscopic only, volumetric, and perspective displays were 20.4%, 20.4% and 27.7% respectively. Pair-wise means comparisons using Tukey's post-hoc test showed that the stereoscopic display with head tracking was significantly better than the volumetric display, which in turn outperformed the perspective display ($p<.05$) (Figure 3-8).

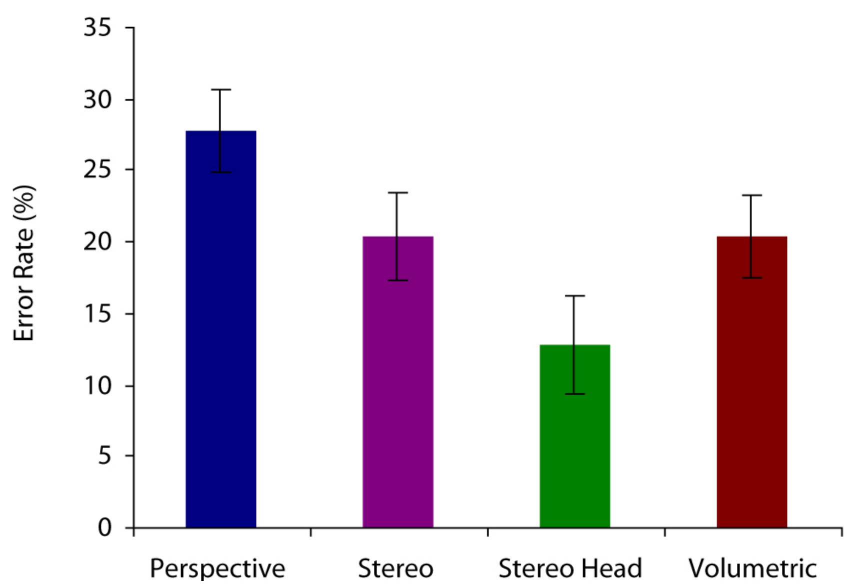

Figure 3-8. Error rates for the path tracing task.

Our analysis of graph orientation found that in 6 graphs, there was a significant interaction between orientation and display ($p<.05$ in all 6 cases). Further analysis revealed two causes for this interaction. Firstly, error rates for the perspective display could be quite high for some graph orientations, whereas, on the other displays, the error rates were uniformly low for those particular orientations. By looking at the conditions where this occurred, it was clear that in the orientations where the perspective display performed poorly, occlusions were present in the 2D projection of the 3D graph, making the task especially difficult. In the other displays, users could either use the stereo cues or head tracking to have a better understanding of the graph in the occluded locations. Secondly, error rates for the volumetric display were high in some graph orientations, while the other displays had uniformly low errors rates for these orientations. Examining these cases, our earlier concern about some regions of the volumetric display being harder to perceive was confirmed. In particular, high error rates occurred when important graph nodes or edges were centered at the back of the display furthest away from the user's position. Figure 3-9 shows the error rates by orientation for two of the graphs which had significant interaction between orientation and display.

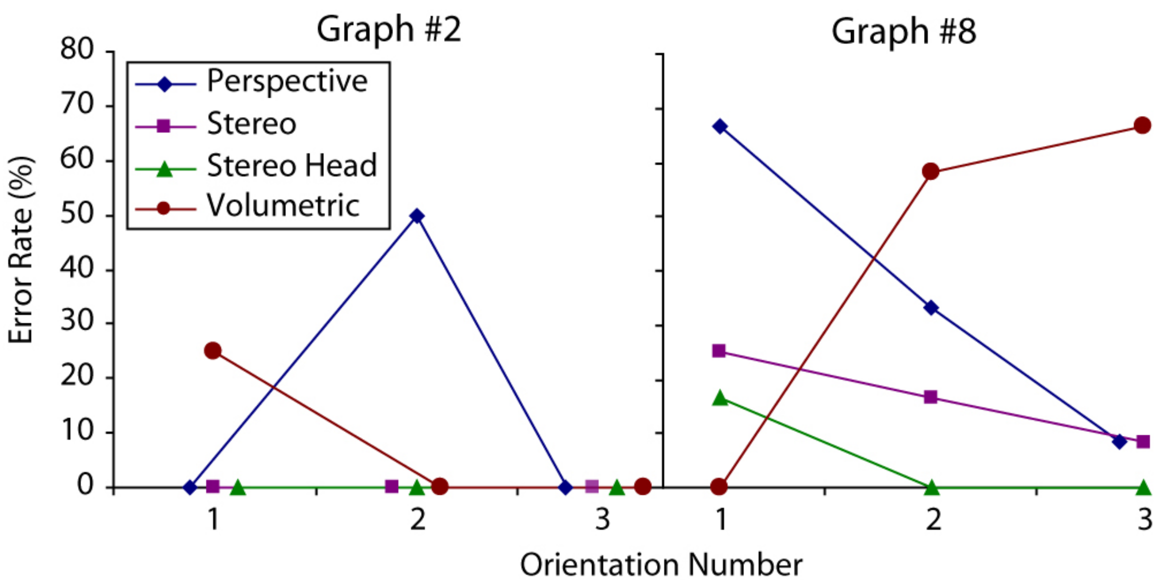

Figure 3-9. Interaction between display technique and graph orientation. Poor results for the perspective display are likely due to occlusions in the graph's 2D projection. Poor results for the volumetric display are likely due to display artifacts.

When we reanalyzed the data with the conditions where goal nodes were in a dead spot of the volumetric display, the average error rate for the volumetric display was reduced to 16.4%. This is still slightly higher than the error rate for the stereoscopic display with head tracking, but the difference is no longer statistically significant.

3.3.6.3 Task 3: Potential Collision Judgment

As with task 2, the main performance measure for this task was the error rate, defined as the average number of errors per trial. Analysis of variance showed that display type had a significant effect on average error rate grouped by condition ($F_{3,33}$ = 39.50, p<.0001). As in the depth rating task, the volumetric display provided the best result, with an error rate of 19.3%. The error rates for the stereoscopic display with head tracking, stereoscopic only display, and perspective display were 22.4%, 27.9% and 48.3% respectively (Figure 3-10). These error rates indicate that none of the display platforms can be used to complete the task with a high level of accuracy. However, the results can still be used as a metric of comparison between the display modes, which is our main motivation. Pair-wise means comparisons using Tukey's post-hoc test showed that the difference between the volumetric display and the stereoscopic display with head tracking was not significant, but the volumetric display had a significantly lower error rate than the stereoscopic only display (p <.05). The high error rate for the perspective display was significantly different from all other displays (p<.0001) and its value, which was close to 50%, shows that the participants were basically performing at the chance when stereo and motion cues were not available.

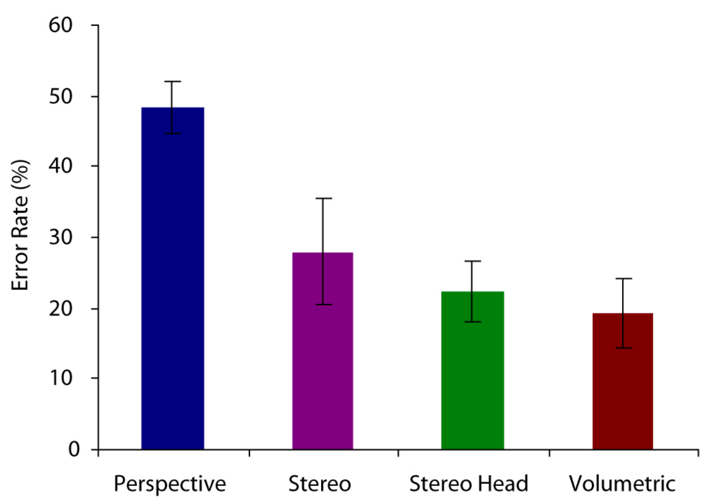

Figure 3-10. Error rates for potential collision judgment task.

3.3.6.4 Head Movement Analysis

We analyzed the head movement data for the stereoscopic display with head tracking and volumetric display conditions. This data was obtained from the head tracker which was used with these display conditions. Because viewing durations were different for the three tasks, we measured the amount of head movement in inches per second, allowing us to compare head movements across tasks. The head movement measure was thus obtained by dividing the total head movement by the viewing duration, for each trial.

For the depth rating task, the volumetric display resulted in significantly more head movement than the stereoscopic display with head tracking ($F_{1,11} = 26.70$, $p<.0001$). This was also the case for the path tracing task ($F_{1,11} = 93.39$, $p<.0001$). For the collision judgment task, the rank ordering was reversed, with slightly more head movement observed for the stereoscopic display with head tracking ($F_{1,11} = 4.69$, $p = .053$). Figure 3-11 illustrates these results.

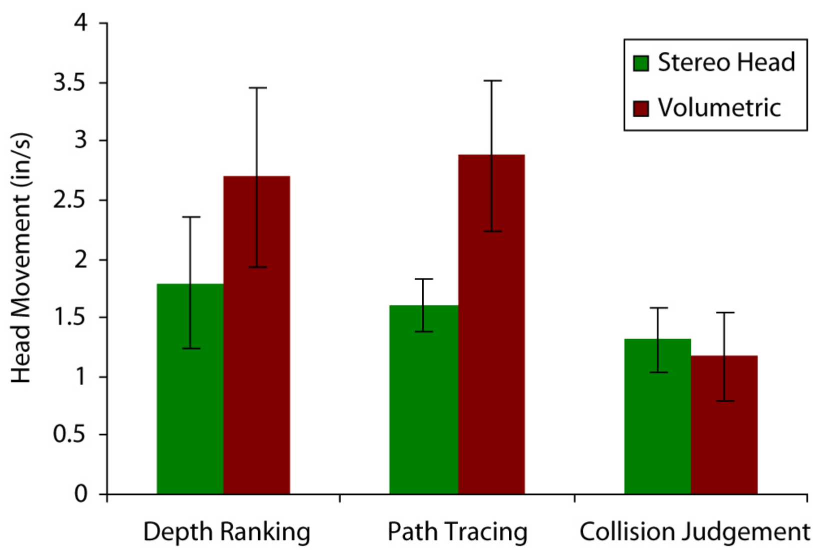

Figure 3-11. Head movement data (in/s) for the stereoscopic display with head tracking, and volumetric display conditions.

As can be seen in Figure 3-11, very similar head movements were used during the path tracing and depth rating tasks. There are a couple of possible reasons why more head movement was used on the volumetric display. Firstly, participants may more naturally use head movements when viewing 3D scenes in true 3D space. Secondly, head movements may have increased for the volumetric display because image quality can depend on the viewpoint.

As for the collision judgment task, both displays resulted in fewer head movements, and the difference between the two displays was much smaller than in the other two tasks.

Head movements may have been used less because it was a dynamic task. Our observations indicated that head movements were mostly vertical in this task, to provide a higher viewing elevation.

3.4 Volumetric Display Artifacts

Volumetric displays offer a number of promising properties which clearly warrant a systematic experimental evaluation of their capabilities. Because the technology is still very new, the display quality is relatively low when compared to the far more mature current generation of 2D and stereoscopic display systems used in this study. This means that there is room for improvements in the quality of the volumetric display hardware, and consequently in the user performance results which we have reported. For example, as the display's quality improves in future generations of the hardware, the significantly better performance for the volumetric display found in tasks 1 and 3 could further increase in magnitude, and for task 2, the volumetric display could outperform the other techniques.

The following is a list of problems in display quality with the volumetric display we used. These may have resulted in less than ideal performances in our reported results:

Brightness: In the display used, the projector brightness is quite low. To be able to see the interior contents of the display, the room lighting must be very low.

Image Stability: The three dimensional images in the display used are generated by a flat projection screen sweeping out the display volume. Slices which are 180° apart are therefore perfect mirror images of each other. These "front" and "back" projections are not perfectly aligned, and as a result, images appear to be shaky in certain areas of the display. The magnitude of the instability is location dependent, and ranges from non-existent to quite severe.

Refresh Rate: At 198 slices per revolution, and 24 revolutions per second, the projector is updating at an enormous rate. However because the entire viewing volume is only being updated at 24 Hz, there is a noticeable flicker in the displayed image.

Faint Spots: With low ambient lighting, most of the viewing volume can be seen quite clearly. However, objects which are in line with the user's eye position and the center of the display are much harder to discern, as the slices which these images are projected onto are parallel with the user's line of vision.

3.5 Discussion

Before conducting our study, volumetric displays were known to have a number of beneficial properties unique to the technology:

Consistent depth information: accommodation and convergence cues are consistent, so users do not suffer from asthenopia.

Minimal hardware requirements: special glasses, head mounted devices, and head tracking technology are unnecessary.

360° viewing angle: Imagery can be viewed from any angle, allowing simultaneous viewing by multiple users.

One factor which was unclear from the literature, which we explored in our study, was whether or not volumetric displays provided superior depth perception. Results of the depth rating task show that volumetric displays indeed have the potential to improve depth perception in comparison to stereoscopic displays, even when the stereo view is coupled with head tracking hardware. However, the results from our second task indicate that the technology needs to improve before the display is ready for actual use.

Because of the artifacts in the volumetric display, we could not ensure uniformity in factors such as brightness, contrast, and resolution across all the display types in the experiment without reducing the viewing quality of the other displays, which would be unfair to those displays. In essence, our experiment biased against the volumetric display, and even so the results are mostly in favor of it, which means it can only do better as the technology improves.

Conversely, for the other display techniques we used a perspective projection that simulated the volumetric display volume. This resulted in a medium-level perspective effect. It may be the case that a stronger perspective projection which went beyond the perspective cues obtained from viewing the volumetric display would result in better performance for these displays.

We also found that more head movements were used when viewing the volumetric display than when using the stereoscopic display with head tracking. Although this was partially a result of users needing alternative views due to volumetric display artifacts, it also implies that users might be more comfortable moving their heads when they are viewing true 3D imagery. It would be interesting to further explore this issue and compare results with head movements used when inspecting physical 3D objects.

Our results should also be interpreted in light of the body of existing work. We intentionally chose three baseline display techniques for comparison which have been found to provide statistically different results in depth perception tasks. The results which we obtained for the baseline conditions match these previous results, with the same continuum being found, in all three tasks. This further validates our experimental method and results, providing strong evidence as to where the volumetric display lies in this continuum.

3.6 Implications for User Interface Design

The most important result which we have obtained is that volumetric displays significantly improved depth perception in our low-level depth raking tasks. This result indicates that volumetric displays have the potential to improve depth perception of three-dimensional imagery. This has important implications to the future development of volumetric display applications. While designers could choose a volumetric display as the platform for an application because of its qualitative properties and unique affordances, they now also have justification to choose volumetric displays as a platform because of quantitative benefits. If users are required to carry out tasks which rely on their perceptual understanding of three-dimensional data, then performing that task on a volumetric display could result in improved performance.

Despite the improved depth perception, the volumetric display did not perform as well in the path tracing task, most likely due to the viewing artifacts associated with the current generation of display which we used. This is an important result, as it shows that the quality of the display must improve before it can be adequately used for real-world 3D tasks. We discussed the viewing artifacts which were present, and when technology advances address these artifacts, we can expect that the users' viewing experiences will likely improve.

3.7 Future Work

While the results of the depth rating task give us indication that volumetric displays *could* be beneficial in three-dimensional tasks, more work is required to determine if the displays *will* be beneficial. To get a completely accurate answer to this question will we need to wait for the technology to advance to a level where the displays can actually be used in real world tasks, and to then study the display in a real usage scenario. However, more work can be done before the technology reaches this level.

For example, our depth rating task tested a user's ability to determine the depth of a single object. While this task provided us with necessary data on the lowest-level issue, follow-up studies could look at higher level tasks. It would also be useful to evaluate the user's ability to perceive relative spatial locations of two or more objects. This would be very important in tasks such as surgical planning, where a surgeon must have a solid understanding of the complex spatial relationships of numerous anatomical structures.

Another evaluation would be to look at the effect that viewing a 3D structure on the volumetric displays has on a user's cognitive model of that structure. This would be an important finding if the display were to be used as an educational tool.

Finally, it would be of interest to study the volumetric display in a task where users have to identify deformations in a three-dimensional structure, or find differences between two structures. Results from such evaluations would have important implications to using the volumetric display as a tool for diagnosis.

For all the above-mentioned studies, it would be interesting to evaluate the volumetric display when users are allowed to walk around the display, instead of having a fixed viewpoint, as they did in our evaluation. It would be interesting to compare such a viewing mode to immersive 3D displays, such as an HMD or CAVE, where the users could also walk around the virtual imagery.

One factor which we did not directly analyze was learning. We omitted blocks from our experiment design to limit the experiment to a reasonable duration, since there were already four display techniques, and three tasks, to perform trials on. As a result, a formal analysis of learning effects was not possible. It may be the case that users will learn to use some display technologies faster than others, and would be observed from a interaction effect between block number and display condition. This is another potential line of future work.

Another interesting line of research would be to evaluate the volumetric display in comparison to viewing true 3D physical objects. For example, a physical replica of the depth rating task could be assembled. Such an evaluation would be interesting as it would reveal how much potential room for improvement there would be in the display technology.

3.8 Conclusions

Our experiment has shown that in addition to their desirable qualitative properties, volumetric displays can improve users' depth perception, making them a potential platform for 3D display applications. That said, these displays are likely unsuitable for widespread use until better performance is observed in complex situations such as in our path tracing task. We imagine that this is only a matter of time, and will occur when technological advances improve the display quality and eliminate the described display artifacts. Furthermore, our studies only looked at low-level and abstract depth perception tasks. We have discussed possible follow-up studies, which may be more representative of actual usage tasks of the volumetric display, but we believe the results from the current study will serve as a baseline for any such follow-ups. As with the study presented in this chapter, such follow-up studies would also be comparative in nature, assessing the volumetric display's performance in comparison to other 3D display forms. Another important line of work, however, is to assess the perceptual issues which are unique to the volumetric displays. In the following chapter, we initiate this line of work.

4. Investigating Text Readability

"You can observe a lot by just watching."

-Yogi Berra

4.1 Introduction

[1]In the previous chapter we investigated the impact that a volumetric display's true three-dimensional display volume has on human perceptual abilities. We discussed a study which showed that volumetric displays provide improved depth perception in comparison to more traditional 3D display platforms. Such a comparative study is important in assessing what, if any, benefits would be gained by developing an application on volumetric displays, in comparison to other 3D display technologies. In this chapter, we continue our line of research into the perceptual human factors associated with volumetric display usage. However, here, we focus on an important perceptual issue which is unique to volumetric displays.

The issue which we focus on is the user's ability to perceive and read text when it is displayed on volumetric displays. This issue is of particular interest to volumetric displays because of its 360° viewing angle. While it is unlikely that volumetric displays will be used for extensive 2D text editing tasks, it is reasonable to expect that textual labels will be used. For example, one could imagine a team of architects viewing and labelling a marked-up 3D model of a new building. With this and other similar scenarios in mind, it is crucial that we be able to display text effectively on volumetric displays.

[1] A preliminary version of this work is published in Grossman et al. [2007].

Because the displays do have a 360° viewing angle, a difficulty with presenting text is that it may not be oriented towards the viewer. This can result in suboptimal performance [Bowman et al. 2003, Chen et al. 2003]. This problem is intensified when multiple users are viewing the 3D imagery from their own unique viewpoints. In this scenario, text which is oriented towards one user may be difficult to read for other users because of their viewing angles.

In this chapter, we present a thorough investigation on the effects of text orientation in volumetric displays. We first present an experiment aimed at quantifying the readability penalty that is incurred when text rendered in a 3D volumetric display is rotated – by a variety of angles about the pitch and yaw axes – such that it is not oriented facing the user. Based on the data from this experiment, we develop a technique for more optimally displaying text on volumetric displays. This technique is validated in a second experiment, where groups of three users simultaneously read text rendered on the volumetric display. Results show that our new technique decreases reading times for groups of three viewers by an average of 33%, making it a useful text rendering technique that could be adopted by designers of volumetric display applications.

4.1.1 Definitions

To aid our discussion of 3D rotation of text, it is useful to first define some terminology. Throughout this chapter, we will refer to the rotation of text as *roll*, *pitch*, or *yaw*, with rotation angles specified in degrees. *Roll* is the rotation of text about the axis perpendicular to the face of the text (or Y-axis), with a positive angle representing a counter clockwise rotation (Figure 4-1b). *Pitch* is the rotation of text about the axis that runs horizontal across the text's face (or X-axis), with a positive pitch meaning the top of the text is rotated forwards (Figure 4-1c). *Yaw* is the rotation of text about the axis that runs vertical across its face (or Z-axis), with a positive yaw being a rotation with the left side of the text rotating forward (Figure 4-1d).

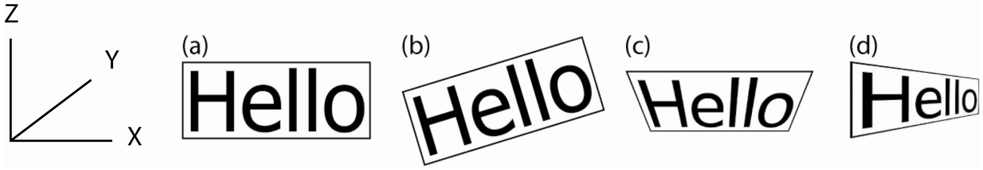

Figure 4-1. (a) No rotation applied. (b) Positive roll. (c) Positive pitch. (d) Positive yaw.

4.2 Related Work

In Section 2.8.4 we provide a thorough review of relevant literature associated with the issue of displaying text for user's which may be at various orientations. Here we summarize those results, and describe how our work will differ.

4.2.1 Displaying Text in 2D Environments

When text is rendered on a 2D display, roll is typically the only orientation variable of interest. Although there are some cases where a single user application may elect to render text rolled away from a traditional 0° orientation [Fitzmaurice et al. 1999], it is primarily within the domain of co-located collaborative systems that orientation of on-screen objects is especially important.

When multiple users share a display, as in the present work, the orientation of information is slightly different for each user, as no two users are seated at precisely the same viewing angle to the display. In the most extreme cases, such as tabletop systems, text can appear upside down to one user while facing another right side up. Wigdor explores the use of orientation on tabletop systems, and experimentally evaluate how text orientation impacts reading speed on tabletop displays [Wigdor and Balakrishnan 2005, Wigdor 2008]. Although informative, their work is limited in its application to 2D environments, since they limit their examination to the effects of roll. In this chapter, we are concerned with the effects of text orientation in a shared, true 3D display environment. As such, an examination of the effect of pitch and yaw on reading speed is essential.

4.2.2 Displaying Text in 3D Environments

Balakrishnan et al. demonstrated interaction scenarios for volumetric displays via wizard-of-oz prototypes [Balakrishnan et al. 2001]. Although they did not explore text display per se, they did discuss strategies for displaying widgets for multiple viewpoints, such as rotating them towards the user, or displaying multiple copies for various viewpoints. While we are unaware of any other investigation into the issue for volumetric displays, text orientation has been explored for other display forms. Larson et al. demonstrated some effects of 3D rotation of text on its readability [Larson et al. 2000]. They found that reading speed was consistent with yaw of up to ±55°, but that orientations beyond this had a significant effect. They also found that the effect was asymmetric: renderings where the first letter appeared smaller than the rest of the text tended to be read slightly slower than those where it was larger.

The present work varies from, and builds upon, Larson et al.'s work in several ways. First, Larson et al. examined the reading of text on a 2D perspective display. The volumetric display, in contrast, will render text in true 3D space, allowing viewers to change their viewpoint of the displayed imagery by moving their heads. Second, because they were concerned only with scenarios likely to occur on 2D displays, the rotations were limited to 70°. The present work demonstrates the effect of rotation about the full compass, necessary since users of volumetric displays can stand anywhere around the display. Third, their study was limited to examining the effect of rotation about the vertical axis (yaw), while the present work examines text rotated about both the vertical (yaw) and horizontal (pitch) axes. Finally, we will also explore the use of multiple reading aids, and develop a technique for optimizing text orientations for multiple viewers on a volumetric display.

Feiner et al. define several methods for displaying abstract information related to a 3D world within windows on a heads-up display [Feiner et al. 1993a]. Unfortunately, their designs are limited to a single user. Bell et al. present an algorithm that dynamically improves the positional layout of text within a 3D environment; however, their solution is also limited to a single user [Bell et al. 2001]. Bowman et al. [2003] and Chen et al. [2003] implemented two methods for rendering text in a 3D scene. With a heads-up display metaphor, text is rendered within the view plane, and thus is always oriented towards the user. In their within-world display metaphor, text is projected onto faces of objects in a 3D scene, thus undergoing pitch, yaw, and roll transformations. An alternative they discuss but do not implement is displaying text within-world but to dynamically rotate it to always face the user. Their experiments found the static within-world display to be detrimental to reading, supporting the hypothesis that the 3D rotation of text will increase reading time. Precisely quantifying the effects of these rotations is the goal of our first experiment.

4.3 Text Rotation and Disambiguation

In volumetric display applications, when text is displayed using a within-world layout, or when multiple viewers are present, text may not be oriented towards a user. Under most usage scenarios, it is unreasonable to expect a user to walk around the display just to read a textual label. While one alternative is to present textual labels using a heads-up display metaphor, this would require the designer to clutter up the display space with multiple labels for each individual user. Thus, it is valuable to consider the scenario where a user is required to read text under a 3D rotation. An impediment to recognizing words under such a condition is the introduction of visual ambiguities. As we see in Figure 4-2a, rotational transformations of the text result in ambiguity to not only the word, but also the individual characters.

While surrounding asymmetric letters could disambiguate rotated text, our preliminary observations showed that the ambiguities can still cause confusion and frustration for the reader. To address this, we experimented with the following techniques to allow the reader to correctly interpret the text.

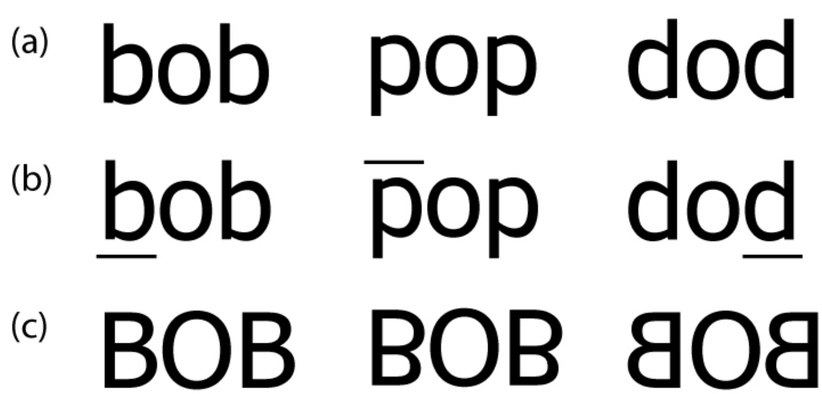

Figure 4-2. The word "bob" under no rotation (left), 180° pitch (middle), and 180° yaw (right). (a) No disambiguation. (b) Underline disambiguation. (c) Uppercase disambiguation.

4.3.1 Underline Disambiguation Technique

In this technique, we employ the relatively lightweight strategy of underlining the first character of the word (Figure 4-2b). This provides disambiguation by identifying both the first character and the bottom of the word.

4.3.2 Uppercase Disambiguation Technique

While the underline disambiguation technique indicates the rotation of the entire word, we also wished to seek out a strategy which could disambiguate individual characters. To do so, we required a typeface in which characters were not ambiguous under rotations. Upper case letters were chosen as a suitable candidate (Figure 4-2c), for which the only possibly ambiguity would be ‘M’ and ‘W’. While displaying text in uppercase can reduce reading speeds [Tinker 1965], we expect the effect to be minimal when reading short textual labels. Furthermore, this technique for reducing character ambiguities is simple and easily implemented. If successful, more subtle approaches to reducing character ambiguities could be developed, such as custom fonts.

4.4 Experiment 1

In this experiment, we seek an understanding of how text orientation in a 3D environment will affect reading times. We wish to quantify the relative effect of a range of rotations and identify particularly problematic ones. We also wish to measure the efficacy of our two disambiguation techniques.

This experiment will be similar in design to the study performed by Wigdor and Balakrishnan, which tested the effect of text orientation on tabletop displays [Wigdor and Balakrishnan 2005]. However, since the text on such displays is inherently 2D, only the effect of roll was examined. In this experiment, we will evaluate the effect of pitch and yaw on reading times.

We can easily predict some results, such as higher reading times when text is upside down or backwards, as well as a significant spike in the reading times at orientations in which the text is parallel to the viewer’s line of sight. Less predictable, however, is at what point, how quickly, and how significantly reading times will increase to these maximal peaks. As for our two disambiguation techniques, we expect that both techniques will decrease reading times in some cases, but we wish to determine under which orientations these techniques are most beneficial, and what the relative efficacy of the two techniques are.

4.4.1 Apparatus

As with our study presented in Chapter 3, we used a volumetric display from Actuality Systems (www.actuality-systems.com). This display is described in detail in Section 1.4. Words were rendered using the default OpenGL Utility Toolkit mono-spaced font. A standard QWERTY keyboard, which was placed directly in front of the participant, was used for text entry. To maximize the image quality of the volumetric display, the room lights were turned off, and a small light was used to illuminate the keyboard. Participants sat in a fixed location chair whose height was adjusted so that all participants viewed the imagery from the same viewpoint regardless of the individual's height. We used a Vicon motion tracking system (www.vicon.com) to track the positions of the viewers' heads. Three passive reflective markers were placed on a hat which participants wore. Six Vicon cameras tracked the 3D location of these markers, and the data was streamed into our experiment application at 120Hz. The application ran on a Pentium 4 PC running at 2 GHz, with imperceptible latency. Figure 4-3 illustrates the apparatus.

Figure 4-3. Hardware apparatus and setup for Experiment 1.

4.4.2 Participants

Twelve unpaid volunteers (3 female, 9 male), ranging in age from 20 to 35, participated in the experiment. Participants were a mix of both undergraduate and graduate students in computer science. We ensured that all participants read English natively, and also had proficient typing skills, when signing up for the experiment.

4.4.3 Procedure

To maximize the general applicability of our results, we abstracted the primary task of interest, reading and comprehending text in a volumetric display, and conducted a basic speed-of-reading test. Participants were presented with a series of common five-letter words which they were asked to read and type into the system. This design was similar to that described in Wigdor and Balakrishnan [2005].

To begin a trial, participants pressed "enter", and the word was presented. We wished to measure the time participants took to read the word before beginning to type. They were, instructed to press "enter" as soon as they recognised the word, which would then disappear, ensuring that they were no longer reading the text. A text box was then displayed, into which they would type the word and then press "enter" to finish the trial. If they typed an incorrect word, a tone would sound and the trial word would appear again, with the trial counted as an error. If, after beginning to enter their response, the participant wished to review the word, they could press "escape" to return to viewing the word – all such trials were also counted as errors. Lastly, if the participant failed to enter the correct word after 30 seconds, the trial would end and be counted as an error.

As outline in Section 3.4 the quality of volumetric displays is relatively low in comparison to 2D displays, as they possess a number of visual artefacts. To ensure this did not affect reading times, we chose a stable and visible location to display the word throughout the experiment. The word was rotated with either a pitch or a yaw. To keep the size of the experiment manageable, we did not vary both pitch and yaw in the same condition. Furthermore we omitted roll rotations, as the effect of such rotations on reading speeds has already been studied in the table top literature [Wigdor and Balakrishnan 2005].

At the start of the experiment, a short calibration program was used to control participants' viewpoints. This was done by adjusting the chair height, and using the motion tracking system to ensure that viewpoints were aligned with the predefined location of the text in the volumetric display. Furthermore, to start any trial, the participant's viewpoint first had to be horizontally and vertically aligned with the center of the text location, with a 1º error allowance. Once the trial began, users were free to move their heads and stand up, as long as their feet did not move.

4.4.4 Design

A repeated measures within-participant design was used. The independent variables were the disambiguation technique, *DT*, the rotation type, *RT,* and the rotation angle, θ. The values for *DT* were *none*, *underline*, *uppercase*, and *both* (both underline and uppercase). The rotation type was either *pitch* or *yaw*. The angle of either the pitch or yaw, θ, took on the 24 values between -180° to 165° in increments of 15°, resulting in 48 text orientations.

The experiment lasted about 90 minutes, and was divided into 4 sessions. Each session consisted of all trials for 1 of the 4 values of *DT*. Each session was divided into 3 blocks, within which the 48 text orientations appeared 2 times each in random order. This resulted in 1152 trials per participant. The five-letter words were chosen from among the most frequent five-letter words appearing in the British National Corpus, with the relative frequency of the set of words chosen for each condition balanced within each participant.

To familiarize participants with the task and disambiguation techniques, four warm-up trials were given. Presentation orders of the disambiguation techniques were counterbalanced using a 4x4 balanced Latin square design. Participants were randomly divided into one of four groups, with each group being assigned one of the four orderings.

4.4.5 Results

4.4.5.1 Reading Time

Reading time is the primary dependent measure, defined as the time between the word appearing and the user pressing "enter" before beginning typing. Recall that once users did this, the text disappeared, ensuring that users had finished reading at this point. In our analysis of reading time, we discarded trials in which errors occurred (5.1% of the data), and removed outliers that were more than 3 standard deviations from the group mean (2.26% of the data).

Repeated measures ANOVA showed main effects for *DT* ($F_{3,33} = 224$, $p < .0001$), *RT* ($F_{1,11} = 581$, $p < .0001$), and θ ($F_{23,253} = 238$, $p < .0001$), with significant interactions between all pairs (all $p < .0001$). Text with pitch rotations were more difficult to read, with reading times of 1.96s for *RT* = *pitch*, and 1.23s for *RT* = *yaw*.

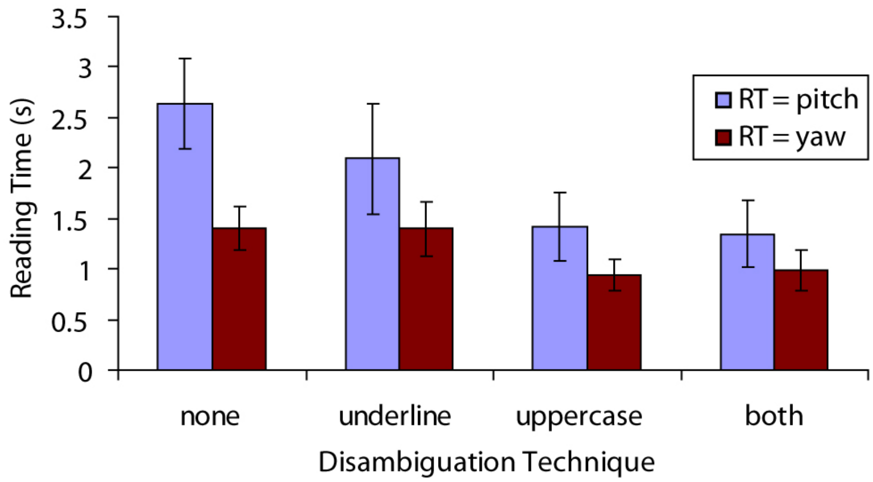

Figure 4-4. Reading times for each disambiguation technique.

DT had average times of 2.06 for *none*, 1.79 for *underline*, 1.20 for *uppercase*, and 1.18 for *both*. Pair-wise means comparison using Tukey's post-hoc test showed that *none*, *underline*, and *uppercase* were all significantly different ($p < .05$), while *uppercase* and *both* were not. The same significant effects were seen for the conditions where *RT* = *pitch,* however for the conditions where *RT* = *yaw*, there was no significant difference between *none* and *underline*, or between *both* and *uppercase* (Figure 4-4). In other words, the effect of the underline disambiguation technique was only significant when the text was pitched, and had no significant effect when the words were already being presented in uppercase.

Figure 4-5 shows the reading times for *RT* = *pitch*. Although reading times seem to increase at pitches of 60° and -60°, post hoc analysis shows that for all values of *DT,* none of the reading times within the [-60°, 60°] interval are significantly different (all p <

.05), showing that pitching the text within this interval has negligible effects. Once outside this interval, the data clearly spikes when the text is close to being parallel with the user's line of sight ($\theta = \pm 90°$). Past this point, reading times drop back down, but since the text remains upside down, reading times are slower than they were within the [-60°, 60°] interval.

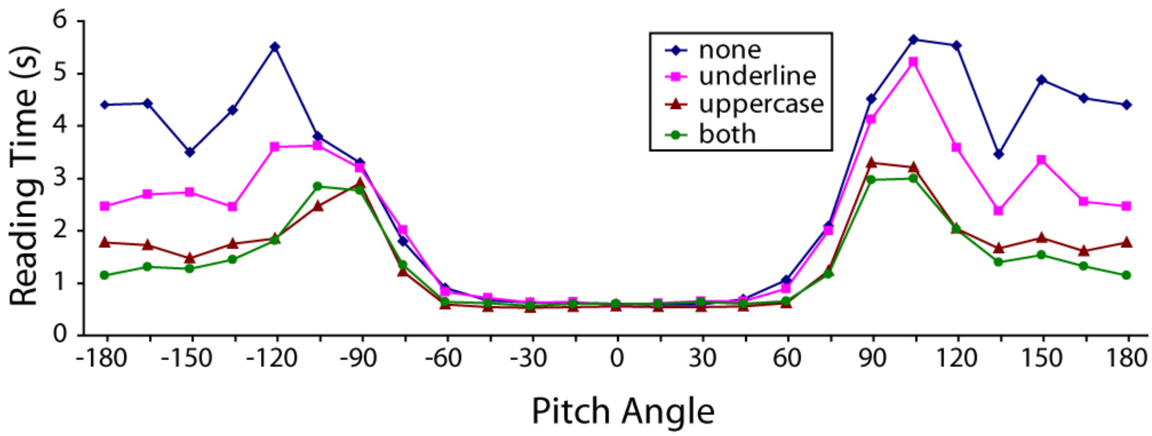

Figure 4-5. Reading times by pitch rotation angle. In Figures 4-5 and 4-6 the data for angle = -180° is repeated at angle = 180°.

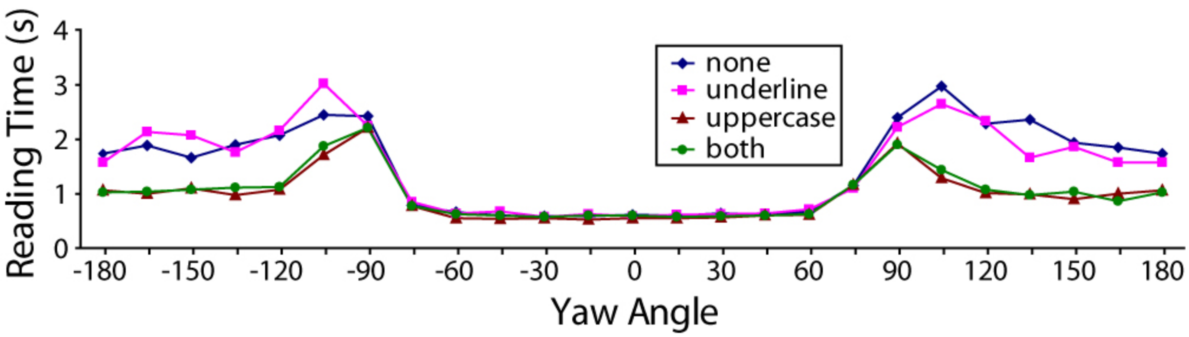

Figure 4-6. Reading times by yaw rotation angle.

In comparison to the condition *DT* = *none*, *DT* = *underline* only reduced reading times for θ = -165°, -135°, -120°, 120°, 150°, 165° and 180°. However, the condition *DT* = *uppercase* reduced reading times for all $\theta \geq 90°$ and all $\theta \leq -105°$. *DT* = *both* produced reading times which were lower than *DT* = *uppercase* outside the interval [-120°, 120°], but this difference was not significant.

Interestingly, this graph is not completely symmetrical. Notably, with *DT* = *none*, pitches of 90° and 105° caused higher reading times than pitches of -90° and -105°. This is likely because it is easier for users to raise their viewpoint rather than to lower it when seated. With -90° and -105° pitches, raising the viewpoint would result in seeing the text right-side-up and forwards, while for pitches of 90° and 105°, raising the viewpoint would result in backwards and upside-down text.

The reading times for *RT* = *yaw* are shown in Figure 4-6. Pair-wise means comparison using Tukey's post-hoc test shows that no data points are significantly different when θ is within the interval of [-75°, 75°]. Note that this interval is slightly larger than the interval for which pitch was not significant. As with the yaw rotations, the reading times spike for θ = -90° and 90°, where the text is parallel to the user's line of sight, and physical movement is required in order to see the text at all. Past this point, the data drops back down but remains slower than within the [-75°, 75°] interval as the text appears backwards.

As for the disambiguation techniques, *underline* only significantly reduced times at θ = 135°. *Uppercase* reduced times significantly for all values of θ outside the interval [-90°, 90°], except for -150°. As with the pitch data, *DT* = *both* provided no further advantage over *DT* = *uppercase*.

Unlike the pitch data, which showed some positive-negative asymmetry in the significant differences, the yaw data is completely symmetric with respect to statistical significance, showing that a yaw rotation has equivalent effects in either direction. We do not claim that this result contradicts earlier results [Larson et al. 2000], since our participants could move their heads to improve their viewpoints. We believe that, unlike with *pitch*, the effect was symmetrical because users could just as easily move their heads left as right.

4.4.5.2 Head Movement

We also measured participant head movement while they were reading text, using the motion tracking system. That head movement was defined as the length of the path the head took during the reading phase of the trial. The effect of text orientation on head movement is similar to the effects on reading time. Head movement was significantly affected by *RT* ($F_{1,11}$ = 218, $p < .0001$), averaging 6.95 inches for *RT* = *yaw* and 11.4 inches for *RT* = *pitch*.

4.5 Orientation Optimization

Our results from Experiment 1 show that even with an unambiguous font, there are still readability problems for text which is rotated in 3D space. Although within-world layouts – where text is rendered on faces of objects in the 3D space – have desirable properties, these results tell us that designers should not use a naïve within-world layout for displaying text within the volumetric display. We now discuss an orientation optimization strategy, which maintains the beneficial properties of within-world layouts, while optimizing reading times for multiple viewers.

The goal for the technique is to present text at the best possible orientation for all users, while still allowing some rotation for within-world display of information. The algorithm searches through possible text orientations, and for each one, estimates reading times based on the orientation of the text relative to each user's viewpoint. The algorithm then chooses the text orientation which minimizes the average estimated reading times across all viewers. It is important to note that this technique does not require our estimated reading times to be precise. Our algorithm only requires the estimates to be positively correlated to actual reading times.

A necessary tool for this technique is a function which, given the position of a user's eye and the relative orientation of text, would provide an estimate of the time required for that user to read the text. To begin, we define $Pitch(\theta)$, $Yaw(\theta)$, and $Roll(\theta)$, which yield reading times for their respective rotation type for an angle of θ. The return values for the *Pitch* and *Yaw* functions are calculated by interpolating over the data points obtained in Experiment 1 (Figure 4-5 and Figure 4-6). Return values for the *Roll* function are calculated by interpolating over the data points reported in Wigdor and Balakrishnan [2005].

If the text's orientation relative to that user's line of sight contained only one of these three rotation types, it would be sufficient to simply use the corresponding function. However, the reading time estimation is more complex, because we now need to consider the case where multiple such rotations exist. For example, if text had 180º rotations of both pitch and roll, the text would not be upside down. It would be incorrect to use the data we obtained in Experiment 1 for pitch = 180º, because those reading times were based on upside down text. Our method is to define viewer axes (V_X, V_Y, V_Z), and text axes (T_X, T_Y, T_Z). The viewer axes are defined as the vector through which the user is looking (V_Y), the up vector perpendicular to this (V_Z), and the cross product of these two vectors (V_X). The text axes are defined as the vector in which the text runs (T_X), the normal vector of the plane on which the text is placed (T_Y), and the cross product of these two vectors (T_Z) (Figure 4-7). If the text is perfectly oriented to the user, then these three axes would all be the same, so we consider the divergence between these three pairs of axes. For example, consider T_X and V_X, with an angle between them of θ. The angle θ is due to a combination of only yaw and roll rotations, as a pitch rotation would not affect V_X or T_X. To estimate the amount of θ which is due to yaw rotation we take the length of the projection of T_X onto V_Y (L_1) and for Roll we take the length of the projection of T_X onto V_Z (L_2). We then take the following weighted average of our reading time functions, $Yaw(\theta)$ and $Roll(\theta)$:

$$Time_X = \frac{Yaw(\theta)L_1 + Roll(\theta)L_2}{L_1 + L_2}$$

We then repeat this calculation for the T_Y - V_Y divergence and T_Z - V_Z divergence. Our final estimation is the average of these three calculated reading times. By taking this average, the reading times for some orientations may be underestimated. Despite this, we expect the estimates to correlate with actual reading times, and therefore serve its purpose in the algorithm.

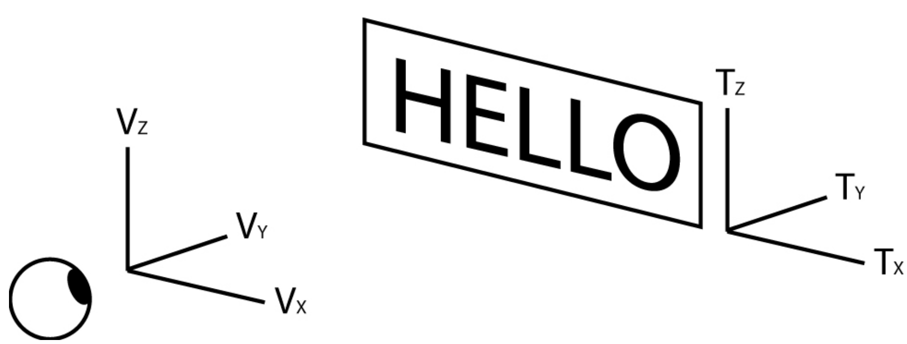

Figure 4-7. Viewer axes (V_X, V_Y, V_Z) and text axes (T_X, T_Y, T_Z) definitions used by the orientation optimization algorithm.

We determine the optimal orientation by searching through all combinations of pitch, yaw, and roll of the text to find the minimum estimated average reading time across all viewers. To allow for consideration of within-world layouts, the algorithm takes an additional parameter, which is the maximum allowed divergence in degrees from the original orientation in any of these axes of rotation. This allows designers to minimize the amount of display volume real estate consumed, and allows users to continue to associate text with the corresponding aspects of the scene. However, even with the maximum divergence defined, we still allow our algorithm to flip the text by 180° along any axis, since the text will continue to be rendered on the same plane, retaining its essential "within world" quality. The running time of this brute force algorithm is thus based on the granularity of the search and the maximum allowed divergence. With sufficiently coarse granularity of the search, this approach runs in real-time.

4.6 Experiment 2

To evaluate the effectiveness of our orientation optimization technique, we conducted a second experiment. Along with testing this new technique, we will make two adjustments to the experimental procedure.

First, since we are testing a technique to aid reading times when multiple viewers are present, we will perform the study with groups of three participants, all reading the same word at the same time. As we have discussed in Section 1.2.3, the volumetric display's 360° viewing angle provide the potential for collaborative use. This study provides an opportunity to validate this claim, with the implementation of an actual collaborative usage scenario. This differs from our own Experiment 1, along with the previous work evaluating text orientations on tabletop displays [Wigdor and Balakrishnan 2005], which were both conducted with a single participant at a time.

Second, we will run the experiment in a more realistic usage scenario, by projecting the words onto faces of a cube. This is an abstraction of a task in which multiple users are viewing a 3D model which is labelled with textual information, using a within-world layout. In such a scenario, the textual labels will rotate in 3D space with the model. As we know from Experiment 1, this will cause reading difficulties at some orientations, justifying the need for our orientation optimization technique.

4.6.1 Apparatus

The same apparatus from Experiment 1 was used. The three users were seated at 120° from each other around the display. Three keyboards were used, one in front of each user, which allowed for text input by all users concurrently. A small flashlight was positioned above each of the keyboards to illuminate them. Figure 4-8 illustrates the setup.

4.6.2 Participants

A total of twelve unpaid volunteers (11 male, 1 female) participated in the experiment. None of the participants had taken part in the first, single user experiment. The experiment was run with 4 groups of 3. Participants were allowed to sign up as groups, and in 2 of the 4 groups, participants knew one another. Participants ranged in age from 19 to 32, and were undergraduate and graduate students. As with the previous experiment, all participants were fluent in English and had proficient typing skills.

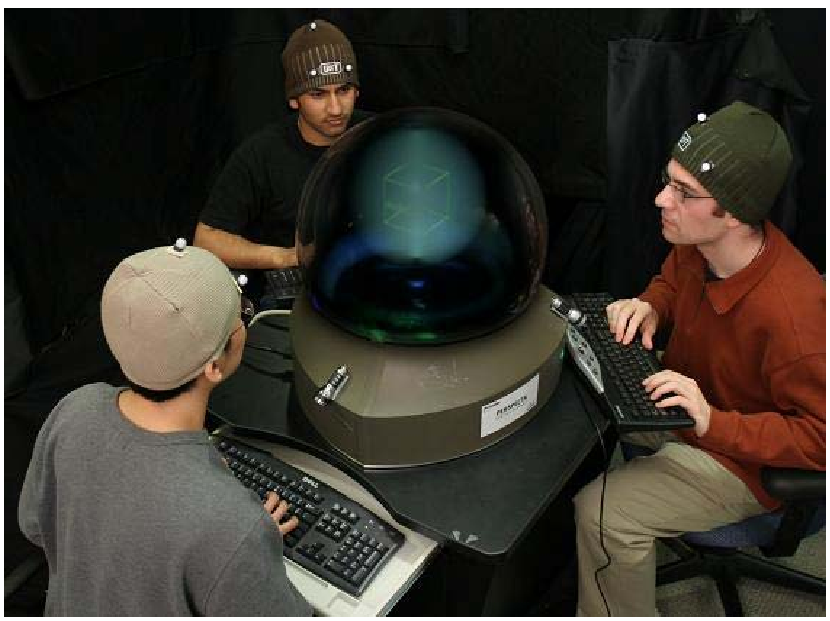

Figure 4-8. Hardware apparatus and setup for Experiment 2.

4.6.3 Procedure

The procedure was similar to that of Experiment 1, with the primary difference being that three participants completed the experiment simultaneously. To begin a trial each user pressed “enter”. Again, we measured the amount of time it took users to read a five-letter

English word before typing it into the system. Once the user had read the word, they would press “enter” again.

Unlike in the first experiment, the word did not disappear at this point, since the other two participants may have still been reading it. Instead, a dense grid was displayed between the participant and the word to indicate that they should no longer read the text. Because it is impossible to render opaque imagery on the current generation of volumetric displays, it was still possible for the participant to “cheat” and read the text. However, participants were under observation, ensuring that they would not do so. When participants pressed “enter”, a text box appeared for them to enter the word. To ensure that participants were not reading from one another’s text boxes, answers were “password hidden”, with characters rendered as ‘*’. Once finished typing, they pressed “enter” again to submit their response. If the input was incorrect, an audio cue would prompt the participant to try again. The trial ended when all three participants submitted the correct response. If a participant had not finished the trial after 20 seconds, the trial would end, being counted as an error for that participant.

Users were seated in three height adjustable chairs directly in front of the volumetric display. The height of each participant’s viewpoint was set to be equal with the center of the volumetric display. Once a trial began, users were allowed the same head movements as in Experiment 1.

A wireframe cube was rendered in the center of the volumetric display. The cube took on various pitch and yaw rotations, and for each trial, one face of the cube was labelled with text. For each cube orientation, there were six trials, one for each face of the cube. Words were displayed in uppercase, as this was the most effective technique in Experiment 1. The underline cue was not used as it provided no effect when words were in uppercase.

The layout of the text was either a naïve within-world projection onto the cube face, or optimized using our orientation optimization algorithm. Although we were capturing the head positions of each user in real time, the three viewpoints used in the algorithm were constant throughout the experiment, based on the location of their chairs, with the height set to the centre of the volumetric display. This allowed us to pre-compute the optimal orientation to a high degree of precision (with a granularity of 1°) using our orientation optimization technique. This simplification also allows us to determine if our algorithm can be effective when precise viewpoint locations are not known. A maximum divergence of 15° was used for the algorithm, which would give it enough freedom to avoid the major spikes seen in Figure 4-5 Figure 4-6. The algorithm used the data from Experiment 1 for *DT* = *uppercase*, since all words were displayed in uppercase for this experiment.

Unlike Experiment 1, it was impossible to design an acceptable experiment in which all three participants were always viewing the text in an ideal area of the volumetric display. As such, we expected to see higher than expected reading times for some trials. While this would have been inappropriate for the first experiment, it was acceptable for the purposes of this experiment, since the goal was no longer to determine average reading times for various orientations, but to test our orientation optimization technique in a real usage scenario.

4.6.4 Design

A repeated measures within-participant design was used. The independent variables were *LAYOUT* (*naïve, optimized*), cube pitch, PITCH (0°, 45°), cube yaw, YAW (0°, 45°), cube face, *FACE* (*left*, *right*, *top*, *bottom*, *front*, *back*), and target user, *USER, (1, 2, 3)*. The naïve layout of the text on each face was determined in a manner such that we obtained a good range of text orientations throughout the experiment. Figure 4-9 shows the naïve text layout and the four possible orientations which the cube took on. The labels in the *optimized* condition diverged from the illustrated naïve layout by a maximum of 15°, as determined by our optimization algorithm. The *USER* variable indicates which user the cube orientation was relative to. When *USER* = 1, the cube orientations seen in Figure 4-9 were relative to the viewpoint of viewer 1. When *USER* = 2, these orientations where relative to the viewpoint of viewer 2. This was accomplished by increasing the cube yaw by 120°. Similarly, when *USER* = 3, the cube yaw was increased by 240°. This design ensured that by the end of the experiment, all users had seen exactly the same set of cube and text orientations. This design resulted in 12 possible cube orientations (2 pitch x 2 yaw x 3 participants).

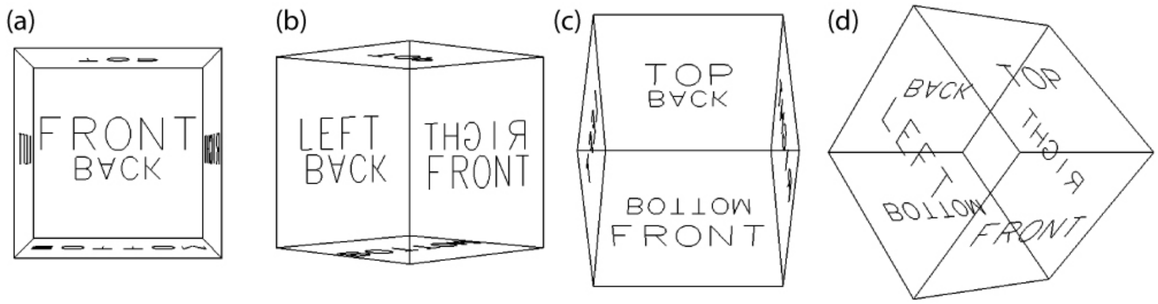

Figure 4-9. Experiment cube orientations, with the naïve text layout. (a) pitch = 0°, yaw *= 0°. (b) pitch = 0°, yaw = 45°. (c) pitch = 45°, yaw = 0°. (d) pitch = 45°, yaw = 45°. These four rotations were repeated relative to each of the three viewers.*

The experiment lasted approximately 70 minutes, and was divided into two sessions. Each session consisted of all trials for one of the two values of *LAYOUT*. In each session the 12 possible cube orientations appeared 4 times in random order. For each cube orientation, 6 trials were completed, one for each face of the cube, also in random order. This design resulted in 576 trials per subject. The same five-letter words which were used in Experiment 1 were used in this experiment, with each group receiving the words in a different order. As in the first experiment, the words were distributed between the 2 sessions such that the average frequency of the words within each session had a similar average frequency in the British National Corpus.

To familiarize participants with the task, six warm-up trials were given before the session began. Ordering of the layouts was counterbalanced across the four groups.

4.6.5 Results

4.6.5.1 Individual Reading Times

We again define reading time as the time to press "enter" before entering the text. Because text did not disappear in this experiment, we removed outliers of typing time (0.9% of data). We also discarded trials in which errors occurred (5.7% of data), and reading time outliers (1.9% of data).

Repeated measures ANOVA showed a main effect for *PITCH* ($F_{1,11} = 199$, $p < .0001$), *LAYOUT* ($F_{1,11} = 465$, $p < .0001$), and *FACE* ($F_{5,55} = 142$, $p < .0001$), but not for *YAW* or *USER*. There was no effect for *USER* since this variable only indicated to which user the cube orientation was relative, so for each value of *USER* the exact same text orientations were seen. There was no significant effect for *YAW* since this variable only defined the cube orientation relative to the target user, and not the orientation of the text.

The most important effect seen here is that the layout had a strong and significant effect. With the naïve layout, reading times averaged 2.13s, and with the optimized layout, reading times averaged 1.42s, a 33% improvement. The layout also showed significant interaction with *FACE* ($F_{5,55} = 39.8$, $p < .0001$), *PITCH* ($F_{1,11} = 6.31$, $p < .05$), and *YAW* ($F_{1,11} = 17.7$, $p < .005$), with the optimized layout lessening the effects of each.

If we combine the variables *PITCH*, *YAW*, *FACE*, and *USER*, we get 72 unique text orientations which each participant saw throughout the experiment. When we look at each of these 72 orientations, in 55 of them the average reading time was lower with the optimized layout. Of the remaining 17, only 3 orientations had significantly slower reading times ($p < .05$). Before concluding that our algorithm failed in these 3 of 72 conditions, we must recall how it worked. The algorithm was designed to optimize the *average* reading time for all viewers for any given trial. So in some cases, the algorithm may increase an *individual's* reading time, in order to reduce the other two viewers' reading times, to obtain the best average reading time for the *group*. This was indeed the case for the orientations where the optimized layout was significantly slower.

4.6.5.2 Group Reading Time

Another variable in which we were interested was the average group reading time for each trial. We wished to know if our algorithm successfully reduced this time, since that is what it was designed to do. We numbered the orientations from 1 to 24, for the 24 possible text orientations which the group saw (2 *YAW* x 2 *PITCH* x 6 *FACE*). Each of these orientations were repeated for the 3 values of *USER*. As with individual reading times, *USER* did not significantly effect average reading times, as this only signified which user the orientation was relative to.

Repeated measures of analysis showed a main effect for both the orientation number ($F_{23,69} = 73.8$, $p < .0001$) and the layout ($F_{1,3} = 499.9$, $p < .0001$), with a significant interaction between the two variables ($F_{23,69} = 14.7$, $p < .0001$). Figure 4-10 shows the average reading times for each of the 24 possible orientations. The reading times were faster for the optimized layout, in all of the 24 conditions, and significantly so in 10 of the conditions ($p < .05$).

This result shows that our optimization algorithm performed as intended, reducing average reading times over the entire group. Figure 4-10 also includes the estimated average reading times which our algorithm used to optimize the text orientation. By definition, the estimated times with the optimized layout are always lower than that of the naïve layout. As expected, the algorithm underestimated the reading times, however, there is still a visible correlation between the estimated and observed times. Regression analysis gives an r^2 value of 0.85, with *a* and *b* values of 2.45 and -0.67 respectively (Figure 4-11).

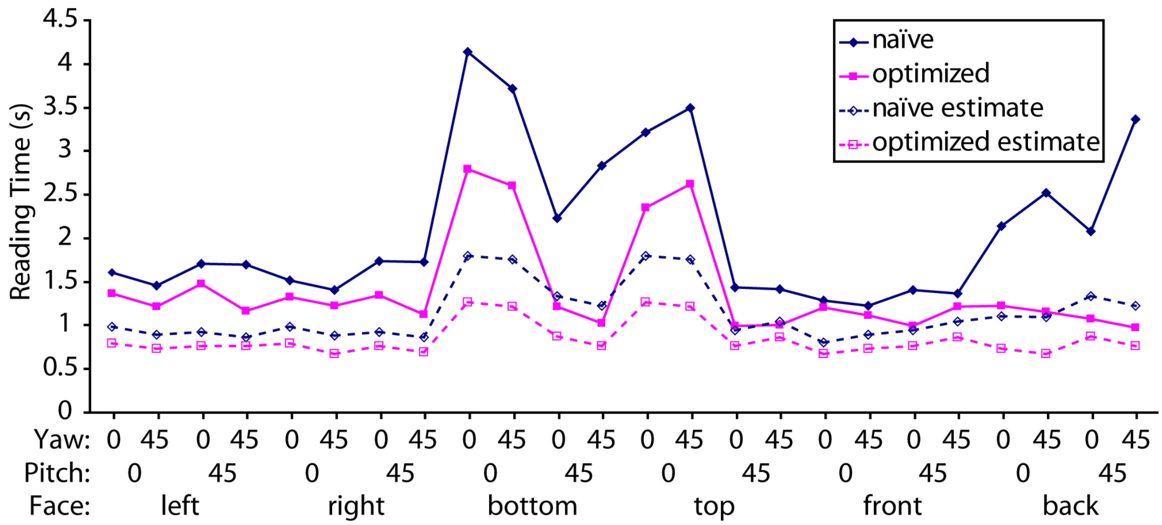

Figure 4-10. Actual and estimated group average reading times for each layout.

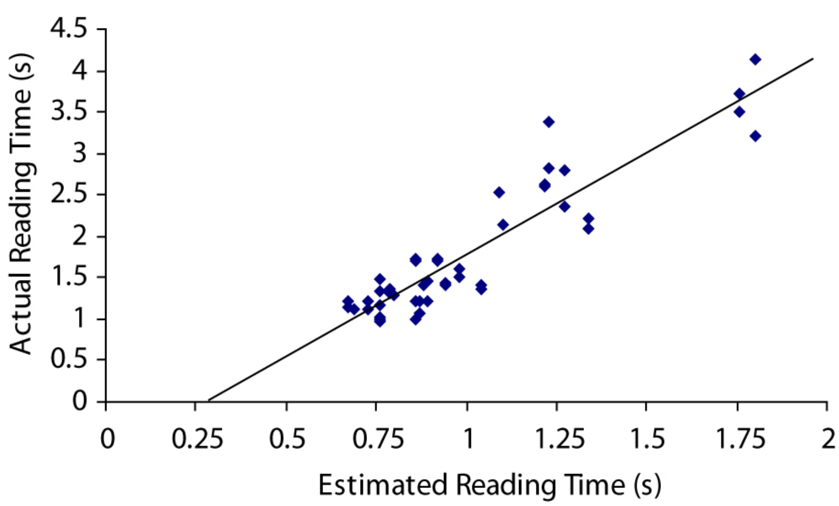

Figure 4-11. Comparison of estimated and actual reading times.

4.6.5.3 Head Movement

As in Experiment 1, we measured the participant's head movement using the motion tracking system, defined as the length of the path which the head took during the reading phase of the trial. Head movements were significantly affected by the text layout ($F_{1,11} = 320$, $p < .0001$), with averages of 14.65 inches for the naïve layout, and 8.15 inches for the optimized layout. This is an important result, as the optimization algorithm reduced the need for head movements. The head movements seen in this experiment were also higher than what was observed in Experiment 1. This explains the higher reading times of similar orientations between the experiments, as the poor quality forced users to move their head to obtain a better viewpoint.

4.7 Discussion

We have presented two experiments exploring the effects of 3D rotations on text readability in true 3D volumetric displays. Our motivation was a scenario in which the users are required to read text that may not be oriented towards them. This can occur either with the presence of multiple users, or with a desire to maintain a within-world layout. Additionally, we discussed potential issues which could increase reading times, and suggested two disambiguation techniques to aid reading, both of which were shown to be beneficial in an initial experiment. Based on the results of that experiment, we developed a new technique to optimize text orientation for multiple viewers. In Experiment 2, we validated this new technique within a collaborative experimental task.

Our findings are consistent with, and extend, the literature. First, we found that text not oriented towards the user can increase reading times [Bowman et al. 2003, Chen et al. 2003]. Similar to the findings in Larson et al., we found that yaw only began to have an impact when the text was rotated by 75° [Larson et al. 2000]. In contrast to this previous work, we found that in an environment allowing head movement, the effect of yaw on reading speed was symmetrical. We believe that this is because, with head movement, users reduce the distortion of the text, and so the effect of first-letter dominance is mitigated.

One of the main contributions of this chapter is the orientation optimization algorithm which we proposed and validated. The algorithm optimizes orientations for any number of viewers, regardless of their relative viewpoints. We conducted an experiment and found that it reduced reading times for 3 viewers by 33%, despite a number of simplifications. Firstly, although the effects of pitch, yaw and roll are likely dependent, we used only independent data for the effects of these rotations in the design of our algorithm. Even so, our algorithm was successful, and produced estimated reading times which had a positive correlation to actual reading times. Another simplification was that the algorithm estimated the viewpoints of the users based on their seating positions. Although using real time head position data may have further improved results, we have shown that our algorithm can be implemented without the need for head tracking technology.

Another advantage of the optimization algorithm was that it significantly reduced participants' head movements. This could be especially important for scenarios in which head movements are inadequate for reducing the orientation of text. For example, if the volumetric display were significantly larger, head movements would not change the viewing angle as significantly for some text. Similarly, in a virtual reality environment, it is much more difficult to navigate to a suitable viewing position [Bowman et al. 2003].

It was interesting to observe in Experiment 2 that the head movements of the users did not interfere with each other. The 120° spacing was enough for the users to move freely as required. However, if there were more users, or the users were closer together, it would again become critical for required head movements to be limited, as the users' head movements could physically interfere with one another.

4.8 Implications for User Interface Design

Both experiments we conducted in this chapter have important implications to user interface design. The first experiment provided us with important information on humans' ability to read text which is not oriented towards them. The results show that a pitch rotation, about the X-axis, is more detrimental to reading than a yaw rotation, about the Z-axis. As such, user interface designers should try to minimize pitch rotations of text in the volumetric display. The results also expose character ambiguity as being an important issue when displaying text. This means that designers should use capital letters, or other unambiguous font sets when displaying text. Finally, the results allow us to determine optimal rotations for readers, which could be used by designers, such as how we used it to design an orientation optimization algorithm.

The results of the second experiment also have direct implications to user interface design for volumetric display. The positive results obtained for our orientation optimization technique indicate that this would be a useful technique to implement in future applications. That being said, there are some potential extensions to the algorithm which we discuss in the next section.

Our work also has implications to the design of collaborative interaction techniques for volumetric displays. The second study was the first implementation of collaborative interaction on a volumetric display. Even though the interface and task were very simple, there were some interesting issues we encountered which will be relevant to future groupware applications developed for volumetric displays. Firstly, because all viewers see the same volume, it is impossible to present data to one user without other users seeing it. This raises privacy issues, especially if one user is viewing sensitive information. One solution which we implemented was to password encode users' answers in the experiment. We also enhanced the system with audio feedback, since visual feedback could be distracting for users for which the feedback did not apply. Although all users heard the same set of sounds, users knew when the sounds were directed to them since it was synchronized with their keystrokes (for example, hearing the successful noise as soon as they press "enter" to submit their answer). Similar observations and solutions have been seen in the tabletop domain [Shoemaker and Inkpen 2001]. We further explore these and other issues regarding collaborative interaction in Chapter 8.

4.9 Future Work

In this chapter, we have gained an understanding of how text orientation affects readability in both single and multiple user volumetric display applications. We have also developed an effective technique for optimizing text orientation to improve reading times for multiple users. However, there are still a number of directions that can be pursued to extend our work. Here we discuss possible extensions to our optimization algorithm, and also discuss alternative methods for presenting text.

4.9.1 Optimization Algorithm Extensions

4.9.1.1 Real-Time Solution

Because our implementation of the search phase of our algorithm used a brute-force technique, applications requiring real-time performance would be forced to limit the granularity of the search, reducing the precision of the results. In our experiment, the angles of the text were pre-computed, but an actual implementation would likely require a more efficient real-time algorithm. It is important to note, though, that even if such an algorithm were employed, our experience suggests that the orientation of text should not be updated every time a user's head position changes, but rather only after large viewpoint changes, in order to limit text movement in the display. As such, the value of a real-time implementation may be limited.

4.9.1.2 Defining Optimal

Another issue of our algorithm which could be explored is how the algorithm optimizes the average reading time when multiple viewers are present. In our implementation, we defined *optimal*, as the orientation which minimized the average reading time across all users. More formally, we define R_i as the reading time under the naïve rotation for each user i, and $R(\theta)_i$ as the reading time for user i, for $i = (1, \ldots, n)$, if the text undergoes a rotation θ. The expected average reading time without the optimization is given by:

$$estimated\_time = \frac{\sum_{i=1}^{n} R_i}{n}$$

We find the optimal reading time by searching for an optimal theta value, which stays within the maximum divergence allowance, k:

$$optimal\_time = \min_{\forall\theta, |\theta| \leq k} \left(\frac{\sum_{i=1}^{n} R(\theta)_i}{n} \right)$$

In some cases, this technique caused certain individual reading times to increase, even though the group reading time decreased. Or, in terms of the above definitions, $R(\theta)_j > R_j$, for some user j. One alternative would be to add a constraint which would disallow any orientation which would increase any of the users' reading times. A more generalized approach would be to optimize a weighted average of the individual reading times. The weight would be determined by system-defined heuristics which estimate the importance of a textual label to each user. So the algorithm would assign weight w_i to user i, and then find θ to optimize the weighted estimated average reading time:

$$optimal\_time = \min_{\forall \theta, |\theta| \leq k} \left(\frac{\sum_{i=1}^{n} w_i R(\theta)_i}{\sum_{i=1}^{n} w_i} \right)$$

Obviously the success of such a technique would require adequate heuristics to define the user weights. However, there are a number of assumptions that would likely be safe for the system to make. For example, if a user created a textual label, they could be assigned a larger weight for that label. As another example, if a user was interacting with an area that had an associated textual label, then the algorithm could increase the weight for that user.

4.9.1.3 Divergence as a Parameter

A last modification to our algorithm which could be considered is incorporating the divergence of the text from the original plane into the weighted average, with more divergence increasing the average. In our implementation, a set maximum divergence, k, was defined, and as long as the divergence stayed within this range, all orientations were considered equally acceptable. If the divergence were incorporated into a weighted average, then the divergence could exceed such a maximum value in situations where this would drastically reduce reading times. Using this technique, the expected optimized reading time would become:

$$optimal\_time = \min_{\forall \theta} \left(\frac{\sum_{i=1}^{n} w_i R(\theta)_i}{\sum_{i=1}^{n} w_i} + f(\theta) \right),$$

where $f(\theta)$ is a positive valued increasing function that determines the importance of keeping the text aligned with its original plane.

4.9.1.4 Optimizing Position and Orientation

Our work compliments the work of Bell et al., who presented an algorithm for optimizing the *position* (as opposed to *orientation* in our work) of textual layouts in 3D [Bell et al. 2001]. An obvious extension would be to combine both algorithms, such that the position and orientation of text were optimized for multiple viewers. This would be especially helpful for volumetric displays, since the results of Experiment 2 suggest that the location of text affects the amount of head movement required to read it clearly.

4.9.2 Text Presentation Extensions

4.9.2.1 Multiple Copies

Another direction is to look at alternative methods for presenting text to multiple viewers. One alternative is to present multiple copies of the text, so that users all have their own textual label which they can read. This could be done in a number of ways. If a within-world layout is to be used, then multiple copies of the text could appear in the same location, oriented in different directions. This technique is similar to placing folded seating assignment cards on a table. By writing the seating assignment on both sides of the folded card, a person would be able to read the label form either side of the table. Such a label could be generalized to have more than two sides. However, as the number of sides increased, so would the visual clutter. So even if there were a large number of users, it would probably be preferable to have at most 3 or 4 orientations for a single label. Another problem with this technique, at least with the current generation of volumetric displays, is that the multiple copies of the text would interfere with the users' ability to read the copy that was facing them. This is because the current generation of volumetric displays cannot hide aspects of the imagery which should be occluded from a user's view.

Another way to provide multiple copies would be to use a heads-up view, where users each had their own label which was directly in front of them, projected on the outer surface of the display. This is similar to the technique used by Feiner et al. [1993a]. However, this technique would also result in visual clutter as the number of labels increased.

4.9.2.2 Curved Text

Instead of providing multiple copies for various viewpoints, the system could slightly curve the presentation of the text. One of the main problems identified in Experiment 1 arises when the text is parallel to a user's line of sight. When completely parallel, the user would not be able to se any of the text. If the text was not planar, and instead projected onto a curved surface, then each user would always be able to see at least a portion of it, regardless of the user's viewing location. If the system knew the location of the users, then the curvature could be adjusted such that it were optimal based on the user viewpoints. One problem with this technique is that for a user looking at text from the side, the end of the word could interfere with reading the beginning of the word, if they were occluding one another.

4.9.2.3 Alternate Font Sets

One last alternative is to look at new font sets. Although uppercase letters worked well, and did not increase reading times when the text was not rotated, new fonts could be developed. Such fonts could allow for both upper and lowercase letters while still reducing character ambiguities and improving reading times when viewed from any orientation. The development of such a font set could be inspired by ambigrams. An ambigram (sometime referred to as an inversion) is a graphical figure that spells out a word not only in its form presented, but also in another direction or orientation [Hofstadter 1979, Kim 1981]. An ambigram which reads the same forwards and backwards would be perfect for volumetric displays, since is would appear as if it were forwards regardless of the side the user was viewing it from. Unfortunately, ambigrams are generally difficult to read because of their complex design, and furthermore, ambigrams are not necessarily available for all possible words.

4.10 Conclusions

In this chapter, we continued our investigation into perceptual human factor issues associated with volumetric displays. Whereas we performed a comparative study of depth perception in Chapter 3, which evaluated volumetric displays in comparison to other display forms, we focused on text readability in this chapter, an issue more central to volumetric displays. We have provided data on the effects of orientation on text readability for volumetric displays, which is important for both single and multiple-user applications, because of the displays 360° viewing angle. Based on this data, we have presented a technique to optimize text orientations, which was shown to successfully reduce reading times when the text is being read by a group of users. In addition, we have discussed a number of alternative techniques for optimizing the text orientation, and for presenting the text, which could be considered in the future. These contributions will be valuable for future user interface designers of volumetric display applications. Indeed, we will incorporate these ideas in our own explorations of user interfaces for volumetric displays, which will be presented later in this thesis. However, before we do so, we will continue our investigation into some of the lower-level interaction issues associated with volumetric displays. In the following chapter, we will turn our attention from perceptual issues to motor capabilities, when interacting in a truly 3D medium.

5. Investigation of Pointing in Three Dimensions

"Ain't no sense worrying about things you got no control over, 'cause if you got no control over them ain't no sense in worrying. And ain't no sense worrying about things you got control over, 'cause if you got control over them, ain't no sense worrying."

- Mickey Rivers

5.1 Introduction

[1]As discussed in Section 1.2, volumetric displays possess a number of interesting and unique properties, such as presenting imagery in true 3D space, and providing a 360° viewing angle. In the previous two chapters we investigated the implications of these properties on human perceptual abilities. In Chapter 3 we discussed a study which showed that volumetric displays provide improved depth perception in comparison to more traditional 3D display platforms. In Chapter 4 we investigated the capabilities of users to read text under various orientations. Along with these perceptual issues, another human factor issue, which arises when imagery is presented by volumetric displays, is the impact on the human motor capabilities. The imagery exists in a truly 3D space, and so the users will need to somehow interact with that space.

At the most basic level, interfaces must enable users to easily select virtual elements in the 3D display space. This will involve some combination of an input device, and a selection technique. However, before making the design choices which will determine the device and technique, it is important to understand what the human motor capabilities will be when interacting in the 3D medium. At the core of the selection operation, is pointing – defining a location in the display area, or in the case of the volumetric display,

[1] A preliminary version of this work is published in Grossman and Balakrishnan [2004].

in the display volume. In 2D graphical user interfaces, the task of selection is well understood, and a number of theoretical models based on the inherently 1D Fitts' Law [Fitts 1954] have been proposed which can be used to both predict and optimize user performance when selecting the graphical elements of a 2D user interface [Accot and Zhai 2003, Grossman and Balakrishnan 2005b, MacKenzie and Buxton 1992]. While many studies have been conducted on selection and manipulation interfaces for 3D environments [Hinckley et al. 1997, Ware and Jessome 1988, Ware and Balakrishnan 1994, Ware and Lowther 1997, Zhai 1995], few have looked at developing and evaluating appropriate predictive theoretical models that describe the underlying human behavior.

In this chapter, we study and model user performance in the most fundamental interaction task – pointing – in the volumetric display. A cursor is rendered in the display volume which the user controls with a six degree-of-freedom input device, using an absolute mapping. We identify various factors that could affect pointing performance in 3D, propose several models that could appropriately characterize such 3D pointing behavior, and conduct a controlled experiment that investigates the effects of the factors identified as well as evaluates the proposed models. By investigating this issue, we hope to not only gain a better understand of pointing in 3D environments, but also be able to make informed decisions about the interaction techniques and user interface designs which we explore in the later chapters of this thesis.

5.2 Related Work

One of the more successful quantitative models in HCI is Fitts' law [Fitts 1954], which is used to model pointing tasks in user interfaces [MacKenzie 1992]. It predicts the time MT to select a target of width W and distance (or amplitude) A from the cursor as:

$$MT = a + b\log_2\left(\frac{A}{W} + 1\right),$$

where a and b are empirically determined constants. The logarithmic term is the index of difficulty (ID) of the task. Fitts' law as originally formulated is a one dimensional model, with the target width W being the only movement constraint. It also assumes that the direction of movement is collinear with this W dimension. Numerous studies have been conducted over the years which amply validate this model for 1D pointing tasks (see [MacKenzie 1992] for a review). However, pointing tasks in most standard interfaces are typically two-dimensional, with targets having both height and width constraints. While not as extensive as the literature in modeling 1D pointing, there has been some research studying the effects of varying the width and height dimensions of 2D targets, as well as varying the angle at which the cursor approaches the target [Accot and Zhai 2003, Crossman 1956, Hoffmann and Sheikh 1994, MacKenzie and Buxton 1992].

In the first study on bivariate pointing in the HCI literature, MacKenzie and Buxton [1992] proposed several different formulas for the index of difficulty for a rectangular target, and found two formulations which correlated highly with their experimental data. Their first formulation considers W to be the dimension of the target in the direction of movement *(W')*. This $ID_{W'}$ model thus reduces the 2D task to a 1D task performed along a line from the start point through the centre of the target. Thus, directional constraints are not captured by this model. Their second formulation, which had the highest correlation with their experimental data, is expressed by:

$$ID_{\min} = \log_2\left(\frac{A}{\min(W,H)} + 1\right),$$

where W and H are the width and height of the target. In this case the target is treated as a two dimensional object. The amplitude, however, is still considered to be a one dimensional scalar. This ID_{min} model has since been used in follow-up work [Murata 1999, Ware and Balakrishnan 1994, Ware and Lowther 1997], and was also proposed apparently independently by Hoffman and Sheikh [1994].

A more recent and thorough study conducted by Accot and Zhai [2003] identifies various problems with the $ID_{W'}$ and ID_{min} formulations. The fundamental problem with the $ID_{W'}$ model is that it completely ignores the directional constraints. With the ID_{min} model, the directional constraint (or height) is considered, but not if it becomes greater than the width. Similarly, the width is not considered if it is greater than the height. Thus, this model does not account for data by, for example, Sheikh and Hoffman [1994] that showed it is harder to acquire a square than a rectangle. Another problem with the ID_{min} model is that it ignores the angle of approach. Lastly, the ID_{min} model allows the width and height factors to be interchanged without changing the index of difficulty. In Accot and Zhai's [2003] study it was shown that this last property does not hold. Figure 5-1 illustrates these limitations.

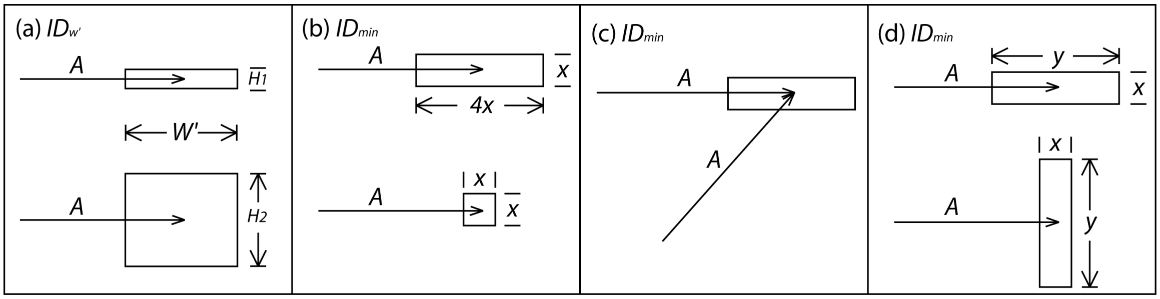

Figure 5-1. Limitations of $ID_{W'}$ and ID_{min} models: all target pairs in this figure are considered identical by the respective ID formulations. (a) No difference in $ID_{W'}$ for targets of different heights. (b) Width has no effect on ID_{min} if greater than height. (c) Movement angle has no effect on ID_{min}. (d) Interchanging width and height has no effect on ID_{min}.

To resolve these problems with existing formulations of *ID*, Accot and Zhai [2003] proposed a number of properties a bivariate pointing model should possess, and developed and experimentally validated a weighted Euclidean model, which we define as ID_{WtEuc}:

$$ID_{WtEuc} = \log_2\left(\sqrt{\left(\frac{A}{W}\right)^2 + \eta\left(\frac{A}{H}\right)^2} + 1\right),$$

where η is empirically determined. This model considers (A/W, A/H) to be the "constraint vector", and they take a weighted norm of this vector and consider it to be the "appropriate distance in a two dimensional space" [Accot and Zhai 2003]. As can be seen, it is similar to the Euclidean norm, with the addition of the parameter η, which allows the model to weight the effect of the height differently from the effect of the width. This ID_{WtEuc} model is a significant improvement over the ID_{min} model in that it alleviates the problem of the larger of the two dimensions not being considered. However, different approach angles and different two-dimensional shapes are still not addressed by this model.

While previous work [Accot and Zhai 2003, Crossman 1956, Hoffmann and Sheikh 1994, MacKenzie and Buxton 1992] has resulted in a good understanding of bivariate pointing in 2D, the same cannot be said for pointing to trivariate targets in 3D. The only work we are aware of which attempts to model trivariate pointing in 3D is by Ware and colleagues [Ware and Balakrishnan 1994, Ware and Lowther 1997] who note that the ID_{min} model can be easily extended to 3D:

$$ID_{\min} = \log_2\left(\frac{A}{\min(W, H, D)} + 1\right)$$

Unfortunately, this model suffers from the same problems as the equivalent 2D model. In addition, this model has only been applied in the context of studies that were primarily concerned with other factors, such as lag and frame rate [Ware and Balakrishnan 1994] and 2D vs. 3D selection [Ware and Lowther 1997], and has not been validated in an explicit experiment that manipulated the three target dimensions *W, H,* and *D.*

5.3 Goals and Directions

The work presented in this chapter is ultimately motivated by the desire to build a sound foundation of theories and empirical data that can guide the development of user interfaces for volumetric displays. This work is one step in this direction, and our goal here is to obtain a thorough understanding of the factors that govern pointing at trivariate targets in 3D, and to develop a predictive model that accurately characterizes such behavior. Our work extends the prior art in three key directions: a new experimental hardware platform, manipulation of experimental parameters, and modeling.

5.3.1 Experimental Hardware Platform

Most studies on 3D manipulation to date have relied on the various stereoscopic displays which we describe in Section 2.2. As we have discussed, such a display's single image plane does not adequately support accommodation, the ability to focus the eye's lens on objects at different depths in the frontal field of view. The investigation which we presented in Chapter 3 demonstrates that this reduces a user's ability to perceive depth. Due to this limitation, previous studies of 3D manipulation have a potentially uncontrolled confound in that participants' perception of the depth dimension is poorer relative to their perception of the other two dimensions.

Unlike stereoscopic displays, volumetric displays allows the human viewer uses their natural physiological mechanisms for depth perception such as true motion parallax and stereopsis through eye convergence and accommodation. Indeed, our results from Chapter 3 show that this can result in improved depth perception. Performing our investigation on the volumetric display will allow us to focus more on the human motor capabilities associated with a pointing task, and worry less about the perceptual constraints.

5.3.2 Manipulation of Experimental Parameters

Target dimensions (W, H, D). It has been shown in bivariate 2D pointing that *H* affects *MT* to a lesser degree than *W* [Accot and Zhai 2003]. We verify if this property holds for trivariate targets, and how the additional parameter *D* affects performance relative to *W* and *H*.

Accot and Zhai [2003] also found that increasing *H* reduced *MT* only when it was smaller than *W*, but increasing *W* reduced *MT* regardless of *H*. In other words, the *W-H* ratio was significant. For trivariate targets in 3D, there are three such interactions which must be examined (*W-H*, *W-D*, *H-D*).

Movement and approach angles. We use the following definition of the three primary axes: the X-axis is the left-right axis, the Y-axis is the up-down axis, and the Z-axis is the forward-backward axis. For the present study, we limit our exploration to physical movements in the XZ plane in 3D space (i.e., Y=0), and explore *movement angles* Θ of 0° (i.e., along the X-axis), 90° (Z-axis), and 45° (XZ-axis). While the movement angle is the human user's axis of movement, the *approach angle* is the angle between the movement vector (defined by Θ) and the axis parallel to the width of the target, as shown in Figure 5-2.

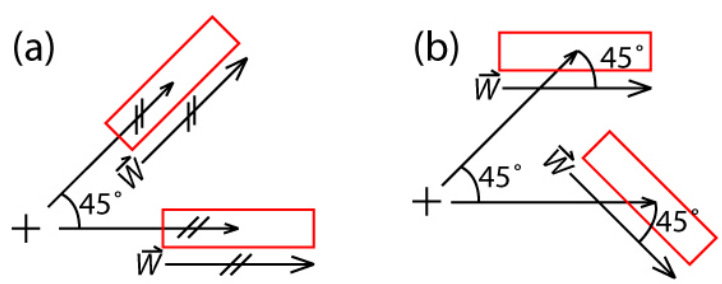

Figure 5-2. (a) Both targets have approach angle of 0° but movement angles of 0° and 45° respectively. (b) Both targets have approach angle 45°.

In previous work [MacKenzie and Buxton 1992], the approach angle was adjusted so that target width was not collinear with the user's movement vector. It is quite probable that this will affect *MT*, and the $ID_{W'}$ model was proposed to compensate for this. However, in altering the approach angle, the physical movement angle Θ was also simultaneously altered. Evidence that Θ affects *MT* has been seen [Boritz et al. 1991], and is partially due to differences in the muscle groups required to affect the different movements.

Because both Θ and approach angle were simultaneously altered in Mackenzie and Buxton [1992], there was no way to positively determine which variable was primarily responsible for the changes in *MT* for the different conditions tested. To eliminate this confound, our study uses a constant approach angle of 0°.

In a trivariate target 3D pointing task, as in 2D [Boritz et al. 1991], it is likely that the changes in Θ will affect *MT* because of the different physical movement requirements. It is also likely that the changes in Θ in the XZ plane will affect the user's perception of the targets due to the targets being at different distances in the visual field. This is likely independent of the quality of the 3D display, since such an effect would occur in the "perceptually ideal" physical 3D world as well. Thus, altering Θ may affect *MT* due to both physical and perceptual differences. We compensate for this possible perceptual difference by counterbalancing the location of targets to both ends of each movement vector.

5.3.3 Modeling

We wish to develop a single model that will accurately capture the various parameters that define user performance when pointing to trivariate targets in 3D.

Despite the deficiencies inherent in the ID_{min} model, for the sake of completeness and continuity with previous work, we include it as the first baseline model in our work:

$$ID_{\min} = \log_2\left(\frac{A}{\min(W, H, D)} + 1\right)$$

Following [Accot and Zhai 2003], we extend this model to accommodate the possible relative effects of different target dimensions, by assigning weights α and β to these parameters, resulting in:

$$ID_{Wt\min} = \log_2\left(\frac{A}{\min(W, \alpha H, \beta D)} + 1\right)$$

We compare this model to a 3D version of Accot and Zhai's [Accot and Zhai 2003] weighted Euclidean model:

$$ID_{WtEuc} = \log_2\left(\sqrt{\left(\frac{A}{W}\right)^2 + \alpha\left(\frac{A}{H}\right)^2 + \beta\left(\frac{A}{D}\right)^2} + 1\right)$$

The problem with these weighted extensions is that, like their corresponding 2D versions, they do not account for possible differences in performance due to varying movement angles. To accommodate movement angles, all components should also be weighted by an additional parameter $f_{W,H,D}(\Theta)$ which takes on different empirically determined values dependent on movement angle Θ. Incorporating $f(\Theta)$ into the ID_{Wtmin} model results in:

$$ID_{Wt\min\theta} = \log_2\left(\frac{A}{\min(f_W(\theta)W, f_H(\theta)H, f_D(\theta)D)} + 1\right)$$

Note that the α and β parameters in ID_{Wtmin} simply get absorbed into the new $f_H(\Theta)$ and $f_D(\Theta)$ parameters in $ID_{Wtmin\Theta}$. Incorporating $f(\Theta)$ in a similar manner into the ID_{WtEuc} model results in:

$$ID_{WtEuc\theta} = \log_2\left(\sqrt{f_W(\theta)\left(\frac{A}{W}\right)^2 + f_H(\theta)\left(\frac{A}{H}\right)^2 + f_D(\theta)\left(\frac{A}{D}\right)^2} + 1\right)$$

These final two models have the same number of free parameters, and differ only in their calculation. The weighted min model only considers one of the three components in the calculation of *ID*, while the weighted Euclidean model incorporates all three components in its calculation of *ID*. With both models, the same extent of "data fitting" will occur, to empirically find optimal values for $f_W(\Theta)$, $f_H(\Theta)$, and $f_D(\Theta)$.

5.4 Experiment

5.4.1 Apparatus

As with the studies presented in Chapters 3 and 4, we used was a 3D volumetric display developed by Actuality Systems (www.actuality-systems.com). The display is fully described in Section 1.4. The display was driven by a 2 GHz Pentium4 computer on which the experimental software ran. The input device used was an Ascension Flock-of-Birds electromagnetic six degree-of-freedom tracker equipped with a single button. The tracker controlled a 3D cursor with a direct one-to-one mapping and a control-display gain of one. The origin was defined to be the center of the volumetric display, with positive X-axis pointing right, Y pointing up, and Z pointing away from the user.

5.4.2 Participants

Five female and seven male paid volunteers participated in the experiment. Participants were screened for adequate stereo vision using the StereoOptical RANDDOT stereopsis test. Participants ranged in ages from 20 to 25, were all right-handed, and controlled the tracking device and consequently the cursor with their right hand. Participants were recruited through flyers and university electronic message boards. None of the participants reported having previous experience viewing or interacting with 3D volumetric displays. Only one of the twelve participants reported having experience with 3D input devices, however this experience was ranked as being “little”.

5.4.3 Procedure

The task was reciprocal 3D target acquisition, which required participants to point to two targets back and forth in succession. The targets were rendered as wireframe cuboids, equidistant from the centre of the display in opposite directions along the given axis of movement. This effectively counterbalanced any differences in perception of targets in different parts of the display. All targets were constrained to the XZ plane (i.e., Y = 0). The target to be selected was yellow, and the other red. When participants clicked on the yellow target, the targets would swap colors, as an indication that the participant had to now move to and select the other target. The 3D cursor that the participant controlled with the handheld tracker was displayed as a crosshair with short line segments along the X, Y, and Z directions all intersecting at its origin. The handheld tracker controlled the 3D cursor with an absolute one-to-one mapping and a control-display ratio of one

5.4.4 Design

A repeated measures within-participant factorial design was used. Objects were modeled in units relative to the radius (5”) of the display volume (1 unit = 5”). The independent variables were amplitude (or distance) *A* (0.44, 0.88, 1.28 units), target height *H,* width *W*, depth *D* (all 0.04, 0.08, 0.16, 0.32 units), and movement angle *Θ* (0°, 90°, 45°) . A fully crossed design resulted in a total of 576 combinations of *A, H, W, D,* and *Θ.*

We use a target-centric definition of *W, H,* and *D* as illustrated in Figure 5-3, where *H* is always measured along the Y-axis, *W* is the dimension of the target along the direction of movement, and *D* is perpendicular to *W* and *H.*

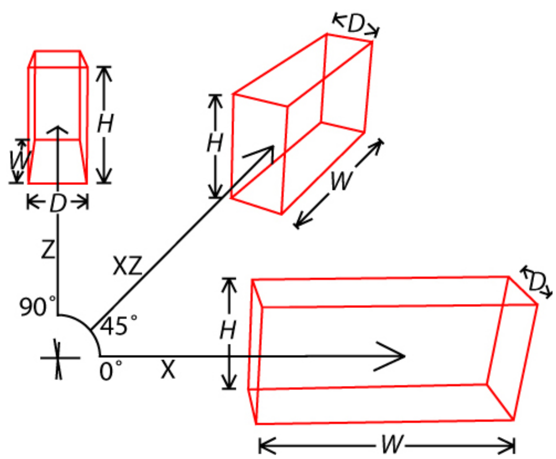

Figure 5-3. Stimulus used in experiment. Targets are constrained to XZ plane and have movement angles of 0°, 45°, and 90°. W *is measured along movement axis,* H *along the Y-axis, and* D *is perpendicular to both.*

The experiment was performed in three sessions, each occurring on separate days. In each session, participants would complete trials for all 192 *H, W, D, A* permutations for one of the three movement angles. For each of the *H, W, D, A* permutations, presented in random order in the session, participants performed a trial set consisting of seven yellow target selections (i.e., six reciprocal movements between the two targets). Because the first selection was used to signal the beginning of the trial set, it required an accurate selection – the 3D cursor needed to be positioned inside the target. Subsequent selections did not enforce this accuracy, but an audible buzzing sound provided feedback when an error was made. After each trial set, the display informed participants of their error rate in the immediately completed set, their cumulative error rate for that session, and the number of trial sets remaining. Participants were asked to minimize their errors. Participants could take breaks between trial sets, but not within each trial set.

Before each session participants were given two practice trial sets to familiarize themselves with selection for the given movement angle.

Participants were randomly divided into 6 groups of 2 each. Assignment of movement angle to groups on each day was counterbalanced using a balanced Latin square. Each session lasted approximately one hour.

5.4.5 Performance Measures

The dependent variables were movement time *MT* – defined as the time between clicks in a trial set, and error rate – defined as the average number of errors per trial. Errors occurred when participants clicked when the cursor was outside the target.

5.4.6 Results

Outliers were removed based on *MT* and accuracy – defined as distance between the click and the target center. Any data point further than 2 standard deviations away from its condition's mean (by *MT*, or by accuracy) was removed. 7.6% of the data were removed as outliers.

5.4.6.1 Movement Time Analysis

Main effects:

Analysis of variance showed that the independent variables W ($F_{3,33} = 2527$, $p < .0001$), H ($F_{3,33} = 715$, $p < .0001$), D ($F_{3,33} = 1577$, $p < .0001$), and Θ ($F_{2,22} = 892$, $p < .0001$) all had a significant main effect on *MT*. Of particular interest is the effect of Θ. Multiple means comparisons showed no significant difference in *MT* for $\Theta = 0°$ and 45°, but a significantly higher *MT* for $\Theta = 90°$. This effect is most likely due to the different muscle groups utilized, as movement for $\Theta = 0°$ and 45° requires limited use of the shoulder muscles, while movement for $\Theta = 90°$ is achieved primarily with the shoulder. This is consistent with the findings of previous studies [Boritz et al. 1991, Card et al. 1991, Langolf et al. 1976] which suggest that the use of smaller muscle groups (hand, wrist) will result in better performance in a pointing task than larger muscle groups (upper arm, shoulder). This effect clearly indicates that movement angle must be accounted for in any attempt to model pointing behavior in 3D.

Effect of movement direction:

Recall that for each Θ we tested (0°, 45°, 90°), two targets were placed equidistant from the origin at opposite directions along the movement axis defined by Θ on the XZ plane. Our reciprocal pointing task required back-and-forth movements in both directions along these three angles. Ideally, one would expect that two equally sized targets on either end of any of these vectors would be equally difficult to select. However, there is a possibility that targets displayed further away from the user in the depth axis are harder to perceive and therefore select. Analysis of variance, however, showed that movement direction did not have a significant effect on *MT,* as Figure 5-4a illustrates. This implies that perception of targets was equally good in all locations.

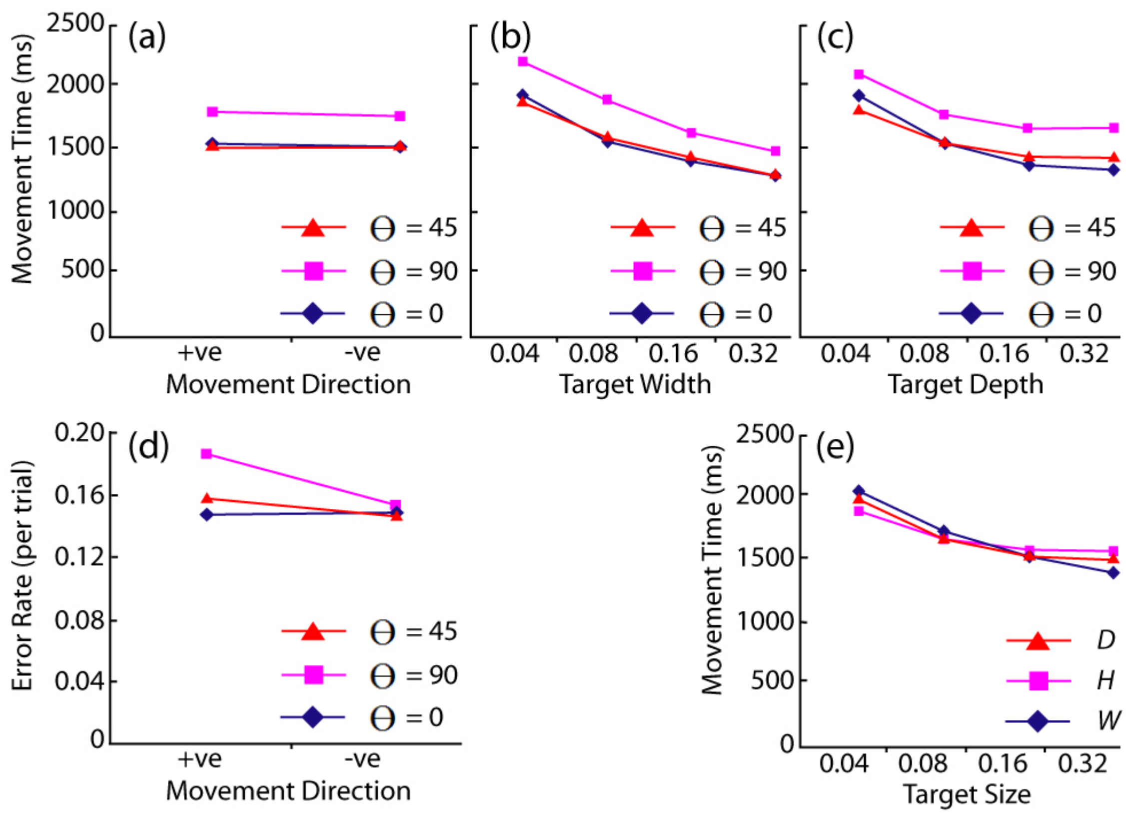

Figure 5-4. (a) Effect of movement direction on MT*, by movement angle. (b) Interaction between* W *and* Θ*. (c) Interaction between* D *and* Θ*. (d) Effect of movement direction on error rate, by movement angle. (e) Relative effect of target dimensions on* MT.

Interactions between Θ and target dimensions:

The effect of Θ on *MT* was dependent on *W* and *D* as indicated by the interactions $W\times\Theta$ ($F_{6,66} = 17.4$, $p < .0001$), and $D\times\Theta$ ($F_{6,66} = 22.5$, $p < .0001$). However, there was no significant $H\times\Theta$ interaction. This is an important result that *likely* occurs because targets were located in the XZ plane and the perception of the height of the target remains constant as Θ changes in this plane. In contrast, the perception of *W* is likely best with $\Theta = 0°$ and worst at $\Theta = 90°$, while the perception of *D* is best with $\Theta = 90°$ and worst at $\Theta = 0°$. Figure 5-4b and c show these interactions.

Relative effect of target dimensions:

One goal of our study was to investigate if any particular target affected *MT* more than the others. Pair-wise means comparison using Tukey's post-hoc test indicated that regardless of the values of *H* and *D*, *MT* significantly decreased when *W* is increased from 0.04 to 0.08, from 0.08 to 0.16, and from 0.16 to 0.32. For *H* and *D*, however, this was only true until the parameter reached a size of 0.16; the increase from 0.16 to 0.32 did not significantly affect *MT.* This is illustrated by the slopes of the lines in Figure 5-4e, where it is clear that the changes in *W* have the most impact on *MT*. This result is consistent with the same effect found in 2D pointing [Accot and Zhai 2003].

Interactions between target dimensions:

Also of interest is whether changes in any one of *W, H,* and *D* would have differing effects on *MT* depending on the size of the other dimensions. Indeed, our analysis showed significant interaction effects for all combinations of dimensions: $H\times W$ ($F_{9,99} = 38.6$, $p < .0001$), $H\times D$ ($F_{9,99} = 17.4$, $p < .0001$), $W\times D$ ($F_{9,99} = 42.7$, $p < .0001$).

An alternative way of looking at this issue is to consider the effect on performance that the ratios between these target dimensions had, which is how Accot and Zhai analyzed bivariate pointing data [Accot and Zhai 2003]. Figure 5-5 shows how the *W/H, W/D,* and *H/D* ratios affect *MT*, broken down by movement angle and the three different amplitudes tested. Note that the ratios 0.125 and 8 represent only single data pairs (e.g., *W/H* ratio of 8 is generated only when W=8 and H=1), whereas the ratios 0.25, 0.5, 1, 2, 4 represent double data pairs (e.g., W/H ratio of 4 is generated for W=8, H=2 and W=4, H=1). As such, to keep the analysis balanced by number of data points, we exclude the 0.125 and 8 ratios from this analysis.

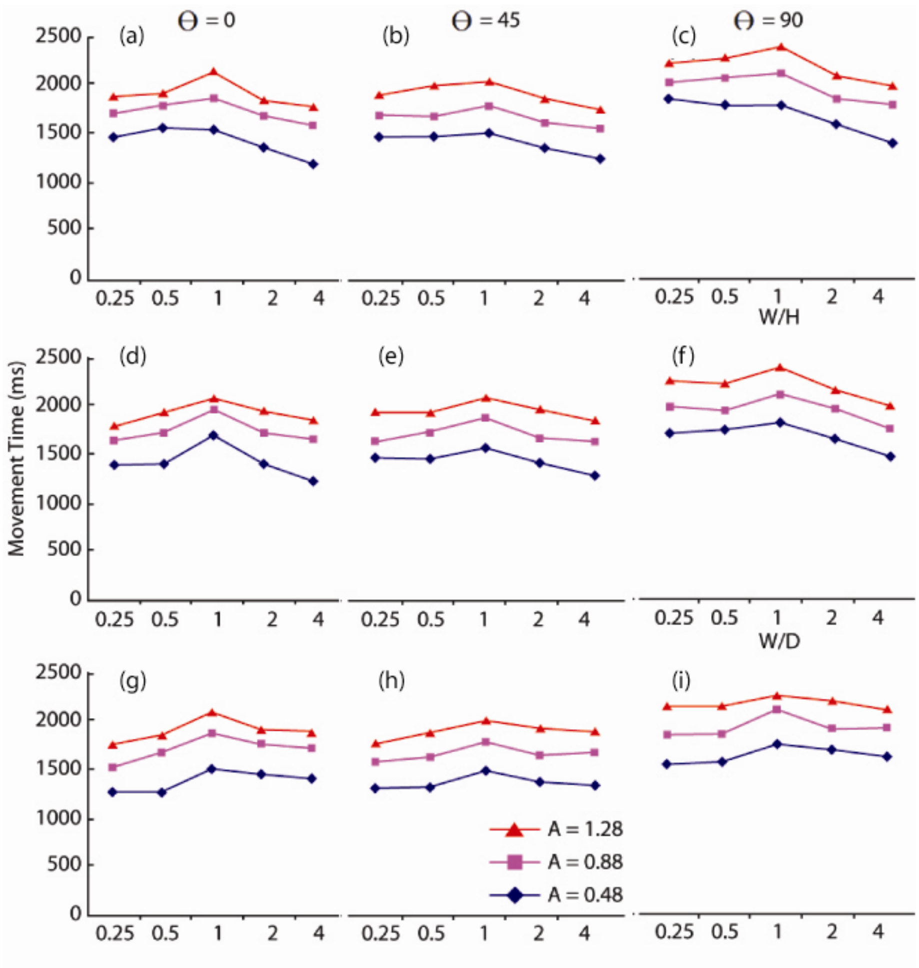

Figure 5-5. (a-c) Effect of W/H ratio on MT. (d-f) W/D ratio. (g-i) H/D ratio.

W/H ratio: There was a significant effect on *MT* ($F_{4,44} = 167$, $p < .0001$), and significant interactions with amplitude *A* ($F_{8,88} = 7.1$, $p < .0001$), movement angle *Θ* ($F_{8,88} = 5.9$, $p < .0001$), and depth *D* ($F_{12,121} = 2.4$, $p < .01$).

For *Θ* = 0° (Figure 5-5a), increasing *W/H* from 1 to 2 had a significant effect on *MT* for all distances. However, increasing it from 2 to 4 significantly affects *MT* only for the shortest amplitude *A* = 0.48. Looking at the left half of the graph, the only significant effects are when *W/H* decreased from 1 to 0.5 for the furthest amplitude *A* = 1.28, and when it is decreased from 1 to 0.25 at *A* = 0.88 (all $p < .01$). Examining the symmetry about unity shows that at *A* = 0.48, *MT* is significantly lower for *W/H* = 2, 4 than *W/H* = 0.5, 0.25. This corresponds to the effect observed by Accot and Zhai. However, when *A*

= 0.88 and 1.28, *MT* for corresponding ratios (0.25 and 4, 0.5 and 2) are not significantly different. Thus, we see that at high amplitudes, increasing *H*, *even when W is the constraining factor*, can still significantly reduce *MT*. While this is in contrast to the results which Accot and Zhai found, they did not break down their results by amplitude, and it is therefore possible that their data exhibited a similar trend for large amplitudes but was simply not reported. As Figure 5-5b and c illustrate, the results for Θ = 45°, and Θ = 90° show similar trends to Θ = 0°.

W/D ratio: There was a significant effect on *MT* ($F_{4,44} = 167$, $p < .0001$), and significant interactions with *A* ($F_{8,88} = 2.9$, $p < .01$) and Θ ($F_{8,88} = 8.3$, $p < .0001$). There was no significant interaction with *H*. Figure 5-5d-f illustrate the trends. The most interesting trend is when Θ = 0° (Figure 5-5d), where the curves are very symmetric about unity. For the two larger amplitudes (*A = 0.88, 1.28)*, the symmetric pairs (*W/D* = 0.5 and 2, 0.25 and 4) do not result in significantly different *MT*, indicating that increasing *D* had the same effect as increasing *W*. However, for *A* = 0.48, *W/D* = 4 results in a significant lower *MT* than its counterpart ratio *W/D* = 0.25. ($p < .001$ in all cases). This result is not observed for the other two movement angles, as might be expected from the interactions between Θ and *D* which indicated that increasing *D* reduces *MT* most for Θ = 0° and less for Θ = 45° and 90° (Figure 5-4c).

H/D ratio: There was a significant effect on *MT* ($F_{4,44} = 111$, $p < .01$), and significant interactions with *A* ($F_{8,88} = 2.0$, $p = 0.5$), Θ ($F_{8,88} = 4.0$, $p < .0005$), and *W* ($F_{12,121} = 3.2$, $p < .001$) (Figure 5-5g-i). Note that symmetry about unity were observed in both Θ = 45° and 90°: for all *A*, *MT* for the symmetric pairs (*H/D* = 0.5 and 2, 0.25 and 4) are not significantly different. However, this isn't true for Θ = 0°, where increasing *D* reduces *MT* more than increasing *H* due to Θ and target dimension interactions (Figure 5-4c).

Fit of the models

We fit the *MT* data to the five candidate models described earlier, using a least-squares method (Table 5-1). Where appropriate, models were fitted by Θ, allowing us to analyze results independent of how Θ effects the time. The last column provides the R^2 values for the regression. The most important values to note from the table are these R^2 values, which indicate the quality of each model.

The poor correlation of the ID_{min} model supports our earlier argument as to its shortcomings. Adding weights to the target dimensions improves the fit, as does including $f(\Theta)$. The ID_{WtEuc} and $ID_{WtEuc\Theta}$ models, however, outperforms all the ID_{min} model variants. The best fit overall, with a correlation of 0.912, was with the $ID_{WtEuc\Theta}$ model where $f(\Theta)$ was calculated independently for each target dimension (note that $f(\Theta)$ was not calculated for the H parameter due to the lack of a significant HxΘ interaction as discussed earlier), and is hence our preferred model:

$$MT \approx 56 + 508\log_2\left((\sqrt{f_W(\theta)\left(\frac{A}{W}\right)^2 + \frac{1}{9.2}\left(\frac{A}{H}\right)^2 + f_D(\theta)\left(\frac{A}{D}\right)^2} + 1 \right)$$

with $f_W(0°) = 0.211$, $f_W(90°) = 0.717$, $f_W(45°) = 0.242$, $f_D(0°) = 0.194$, $f_D(90°) = 0.312$, and $f_D(45°) = 0.147$.

Model	Parameter Estimates (with std error indicated below)											
	Θ	a (ms)	b (ms/bit)	α	β	$f_W(O°)$	$f_D(O°)$	$f_W(9O°)$	$f_D(9O°)$	$f_W(45°)$	$f_D(45)$	R^2
ID_{min}	O°	-187 69.6	454 17.4									0.78
	90°	78 68.9	451 17.2									0.78
	45°	-16 53.2	411 13.3									0.83
ID_{Wtmin}	O°	-163 53.4	472 14.1	1.66 0.09	1.16 0.05							0.86
	90°	118 45.7	477 12.3	1.75 0.08	1.39 0.06							0.89
	45°	11 40.5	428 10.7	1.52 0.07	1.28 0.05							0.89
ID_{WtEuc}	O°	-395 48.7	496 11.9	0.37 0.05	0.91 0.10							0.90
	90°	-23 43.4	494 11.2	0.26 0.03	0.42 0.05							0.91
	45°	-149 36.8	445 9.21	0.41 0.04	0.58 0.06							0.92
$ID_{Wtmin\Theta}$	All	79 25.5	467 7.26	2.13 0.76		1.44 0.52	1.67 0.60	0.85 0.31	1.24 0.45	1.44 0.52	1.82 0.65	0.88
$ID_{WtEuc\Theta}$	All	56 21.6	508 6.58	0.11 0.06		0.21 0.12	0.19 0.11	0.72 0.43	0.31 0.15	0.24 0.14	0.15 0.04	0.91

Table 5-1. Summary of model fitting results. $f_H(\Theta)$ was not calculated since there was no significant $Hx\Theta$ interaction. α and β are calculated instead of $f_W(\Theta)$ and $f_D(\Theta)$ for ID_{Wtmin} and ID_{WtEuc} when presented by Θ = O°,45°, 9O°, because the breakdown of data by Θ makes it unnecessary to compute $f(\Theta)$ for each Θ as is done when data is aggregated across all Θ in $ID_{Wtmin\Theta}$ and $ID_{WtEuc\Theta}$.The last column indicates the effectiveness of each of the models.

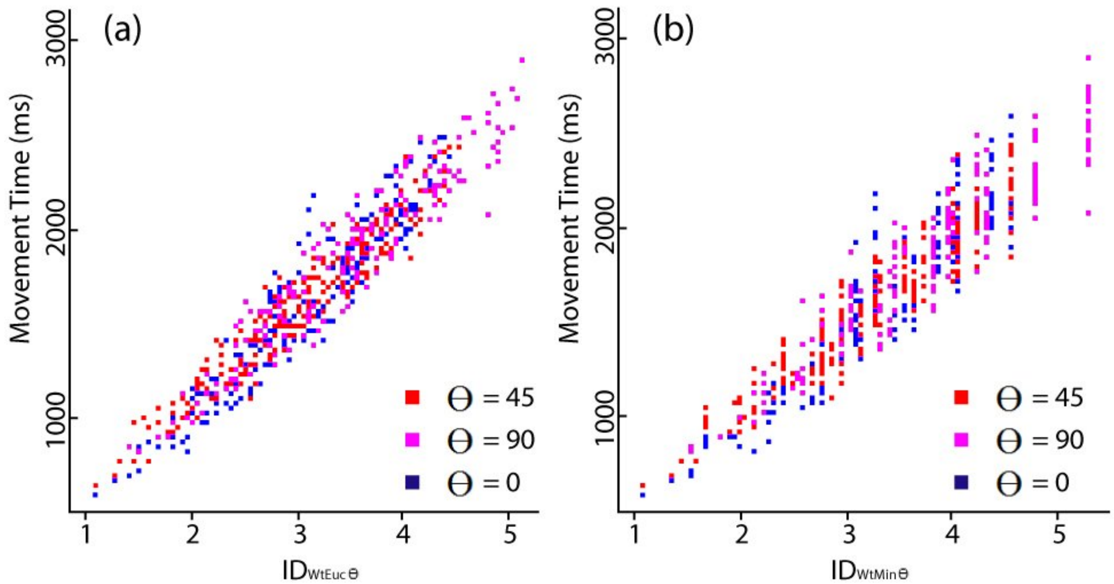

Figure 5-6. Raw data for (a) $ID_{WtEuc\Theta}$ and (b) $ID_{WtMin\Theta}$. The points in the Euclidean model are less spread out, especially as ID increases.

Figure 5-6 illustrates the differences between $ID_{WtEuc\Theta}$ and $ID_{WtMin\Theta}$. The most important effect illustrated by this figure is that the points in the Euclidean model are less spread out, especially as *ID* increases. This demonstrates that the Euclidean model calculates index of difficulties which have a stronger correlation to movement times.

5.4.6.2 Error Analysis

Error rate was significantly affected by *H* ($F_{3,33} = 206.2$, $p < .0001$), *W* ($F_{3,33} = 467.9$, $p < .0001$), *D* ($F_{3,33} = 246.9$, $p < .0001$), and Θ ($F_{2,22} = 13.47$, $p < .0001$). There was no significant effect for *A*. The overall mean error rate was 15.7%. While this is higher than the error rate of 4% seen in typical Fitts' Law experiments, our observations of the participants' behaviors rule out the possibility of subjects "racing through the experiment". Rather, our analysis indicates that when any of the target dimensions *W, H,* or *D* takes on the smallest value of 0.04 units, the error rate goes up significantly ($p < .0001$). When this smallest dimension is not considered, the error rate decreases to an average of 5.4%, which is in line with results reported in the literature. We also believe that secondary sources of error were the greater perceptual and motor difficulty of the 3D task, and noise in our 3D tracker.

Finally, we looked at how the direction of movement affected error rate. Recall that for all three movement angles tested, the direction of movement did not have a significant effect on *MT*. With error rate, however, there was a significant interaction between movement direction and Θ ($F_{2,22} = 7.12$, $p < .005$), and multiple means comparisons indicated that the error rate for positive direction movements when $\Theta = 90°$ was significantly higher than for $\Theta = 0°$ or $\Theta = 45°$ (Figure 5-4d). This effect is likely due slight difficulty in perceiving targets at the far back of the display. However, recall that we did not find a similar effect on *MT* (Figure 5-4a). This indicates that users may not actually think they have perception difficulties with these targets, believing they are in the target and thus completing the task without compromising *MT* but unwittingly making an erroneous selection.

5.5 Discussion

The motivation of the work presented in this chapter was to investigate human motor control capabilities when interacting in a true three-dimensional volume. We focused on pointing in 3D as it is an elemental motor task, which serves as the basis for many interaction techniques, and in particular, selection. We developed a new model for pointing in 3D, and the model was verified by our experimental results. The results which we obtained have important implications to user interface design which we discuss in the next section.

While our study was conducted on a volumetric display, we believe the models we developed would hold on other 3D display platforms as well. However, the constants in the model would probably change. Most notably, the human's perceptual capabilities would change with each display form, as demonstrated by our study in Chapter 3. This means that the dimension parallel to the user's line of sight would become more or less critical, and the associated constants in the model would need to adapt to this difference.

Our work is consistent with the previous literature in 2D pointing. Most notably, similar to the study by Accot and Zhai [2003] we found that a Euclidean model provided more accurate predictions of movement times than the traditional min model. Furthermore, like previous studies in 2D, we found the movement direction to have a significant impact on acquisition times [Boritz et al. 1991].

Our work is also limited in a number of ways which are worth discussing. For one, we only tested movement angles parallel with the ground plane. This was done to limit the experiment to a manageable size and duration. We expect that if vertical movements were required for acquisition, it would further affect movement times. Furthermore, if such acquisition movements were required, the height of the target would like become a more critical dimension, as it would be the constraining dimensions relative to the direction of movement.

A limitation of our model is that there are a number of free parameters which are required to calculate the index of difficulties. With the introduction of parameters, the likelihood of finding data which tightly fits the model increases. On the other hand, introducing these parameters allowed us to define a model which can be generalized to various movement angles. Addressing this issue is one of a number of future lines of work which we discuss in Section 5.7.

5.6 Implications for User Interface Design

A primary motivation of our work was to provide empirical results and practical models that could guide the development of user interfaces for 3D displays. Our study investigated how target dimensions and movement angles affect selection performance, and the results can provide us with significant guidelines about the layout of selectable targets, i.e. how items such as widgets, menus, and other objects should be sized and positioned in 3D displays. The results also provide guidelines for the development of interaction techniques. We now discuss such guidelines.

5.6.1 Guidelines Related to Target Sizing

As in any interface, widgets should be large enough so that they can be easily selected and manipulated, but their size should be minimized to reduce the amount of visual space which they take up. An observation from our experiment is that error rates significantly spiked after they were reduced to a certain size (1/5 of an inch), so clearly targets should not be made this small. More interesting is that designers should avoid targets with grossly disproportionate dimensions, since our results show that having one dimension more than four times as large as another does not make it any easier for the user to select the target. Another important finding was that the dimension parallel to the line of approach (defined as W in our experiment) was always the most critical one. Thus if a target was at the back of the display, it would be beneficial for it to have a larger size in the Z-axis, since the user would most likely be moving forward to select it. Similarly if it were at the side of the display, a larger size along the X-axis would be beneficial.

5.6.2 Guidelines Related to Target Positioning

In typical 2D interfaces, the user's workspace is in the center of the screen taking up most of the display space, while the interface widgets border the screen. Extending this layout to a 3D display could have the workspace in the center volume of the display, and the interface elements on its outside surface. However, our experiment showed that moving forwards and backwards to select targets was significantly slower than moving left and right. If targets were to be positioned on the outside surface, they should be limited to the left and right sides of the surface, maximizing left and right selections.

5.6.3 Guidelines Related to User Locations

It is important to recall that our results, which provide the above guidelines to both target sizing and positioning, were dependent on the location of the user's viewpoint. This presents a challenge for interface designers, since one benefit of volumetric displays is that users can walk around it for full 360° viewing. It would therefore be quite useful if a 3D volumetric display system were able to track the position of its users and then adjust the layout of the interface elements accordingly. We will explore this concept of user tracking further in Chapter 8.

5.6.4 Guidelines Related to New Interaction Techniques

Along with providing guidelines to the design of user interfaces, the results from our study can provide guidelines to the design of new interaction techniques. While it is fair to assume that user interface widgets can be made large enough such that they are easily selectable, elements of the displayed imagery could be arbitrarily small. This could be problematic if the dimensions of such selection targets drop below the critical 1/5 of an inch threshold which we identified. Furthermore, our study shows that even when the objects are larger, the depth of the object still adds additional constraints to the selection task. So unlike in a two-dimensional environment, the task of selection is constrained by three dimensions. As such, new selection techniques should be considered, which reduce the motor constraints on behalf of the user. This can be done in a number of ways.

One option is to increase the motor space activation area of a target, which is independent of a visual size. Such efforts have been explored in 2D user interfaces, and include expanding targets [McGuffin and Balakrishnan 2002], area cursors [Kabbash and Buxton 1995], bubble cursors [Grossman and Balakrishnan 2005a], and adaptive control gain ratios [Blanch et al. 2004].

The alternative is to decrease the number of dimensional constraints associated with the task. A well known technique in the virtual reality literature is the ray casting metaphor [Liang and Green 1994], where the user points a virtual ray at the target of interest. This decreases the degrees of freedom of the selection task from three to two, and should thus reduce the constraints involved in selection which were observed in our study.

5.7 Future Work

The work in this chapter remains to be extended in a number of important ways. We obtained data from pointing in 3D environments, when movements were parallel to the ground plane. It would be useful to acquire data for other movements as well, and to incorporate the data into our models. Furthermore, the models depend on parameters which are functions of movement angles. We found discrete values for these functions, at the three specific movement angles which we tested. A more thorough study looking at movement angles in 3D could possibly determine closed forms of these functions.

Furthermore, our "best-fit" model is somewhat limited in the sense that there are number of parameters which must be optimized by empirically captured data. While this model did prove to be quite accurate, it would be useful to seek out a model with less free parameters. One possible alternative is to extend Grossman's probabilistic model [Grossman and Balakrishnan 2005b] to 3D pointing. This model calculates index of difficulty values by determining probabilities of hitting targets. Although it was developed for 2D pointing, it can be directly generalized to 3D environments.

Finally, while our motivation was to investigate human motor abilities when working in the true 3D volume of the volumetric display, we focused our study on pointing. Further studies could be conducted on other tasks, such as steering [Accot and Zhai 1997], tracking [Zhai et al. 1994], and six degree-of-freedom manipulations [Ware and Jessome 1988]. While these issues remain unexplored in volumetric displays, our work in an elemental 3D pointing task, which is an element for these other tasks, should serve as a basis for hypothesizing results and forming models for the more complex motor tasks.

5.8 Conclusions

In this chapter we have shifted our attention from perceptual human factors, to human factors associated with motor capabilities. We focused on pointing behavior, as a user's ability to indicate a location inside the volumetric display will be essential for the task of selection. Since selection is a core operation in almost any user interface, we were motivated to grasp a sound understanding of the associated user capabilities.

To do so, we presented experimental work that investigated how target dimensions, movement angles, and their interactions effect selection performance in a 3D environment. Unlike previous experiments involving 3D tasks, ours was performed on our 3D volumetric display platform, which provided users with reliable perception of all three spatial dimensions. We observed that moving forwards and backwards in depth is slower than moving left and right for selecting targets; that the target width was more critical than the height and depth of the target; and that the effect of the width and depth were dependent on the movement angle, while the effect of the height was constant regardless of the movement angle. Building upon previous work on univariate and bivariate pointing tasks, we introduced and validated a variant of Fitts' law that models pointing to trivariate targets in 3D. Unlike the previous models, our model also accounts for varying movement angles. We have shown that performance time can be predicted from this model to a greater extent than alternative models which have been previously suggested. Finally, we discussed the implications of our results for the sizing and positioning of user interface widgets in volumetric displays, and also discussed guidelines for the development of new selection techniques. One recommendation which we made is to consider a ray casting metaphor for selection. In the following chapter, we explore this line of work, through the design and evaluation of new selection techniques for volumetric displays.

6. Design and Evaluation of Selection Techniques

"I became a good pitcher when I stopped trying to make them miss the ball and started trying to make them hit it."

\- Sandy Koufax

6.1 Introduction

[1][2]In the previous chapter, we provided an in-depth study of how humans behave when pointing at 3D targets in a volumetric display. The goal of the study was to obtain a solid understanding of the theoretical underpinnings associated with object selection, and to also guide the design of user interfaces and interaction techniques for volumetric displays. Investigating issues associated with object selection is important because it is one of the most fundamental interface tasks for any application, and in particular, has been identified as being a universal task in 3D environments [Bowman and Hodges 1999, Mine 1995a]. Here we build upon the work presented in the previous chapter; following the guidelines which were presented, we design and then evaluate selection techniques for volumetric displays.

In volumetric displays, targets in 3D space must be selected by users who might be located anywhere around the display, a task that is difficult to achieve with existing 2D or 3D selection techniques that are designed for 2D image plane interaction. Research on selection for 3D virtual reality (VR) environments has introduced two commonly used techniques, which may apply to volumetric displays.

[1] A preliminary version of this work is published in Grossman and Balakrishnan [2006b].

[2] This Chapter is supplemented by Video Figure 6-1, available online at: http://www.dgp.toronto.edu/~tovi/thesis and https://tspace.library.utoronto.ca/handle/1807/9944.

Hand extension techniques, or 3D point cursors, directly map the user's hand to the location of a 3D cursor [Hinckley et al. 1994, Mine 1995a, Poupyrev et al. 1996]. Such was the technique used in the study presented in the previous chapter. However, our study exposed a number of difficulties associated with this technique. Most importantly, the depth of the target imposes both a physical and perceptual constraint, not present in the 2D pointing scenario, which results in increased selection times. As discussed in Section 5.6.4, this provides us with motivation to look at the other main technique used for 3D object selection, ray cursors.

Ray cursors, or aperture based selection techniques, project a virtual ray from the user's hand which is used to intersect and select targets [Forsberg et al. 1996, Liang and Green 1994, Pierce et al. 1997]. Studies have shown that for VR environments the ray cursor results in faster selection times [Bowman et al. 1999]. However, this result may not hold in volumetric displays, since all objects are within close proximity to the user, such that the travel distance required of the point cursor is minimal.

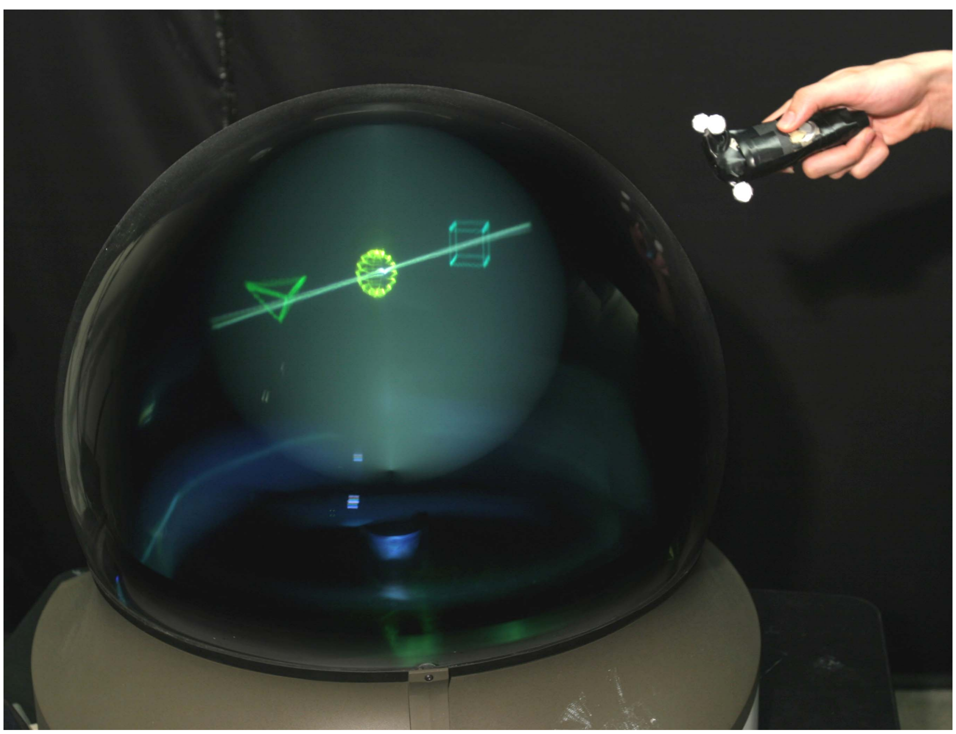

Figure 6-1. Ray cursor selection in a volumetric display. Multiple targets are intersected, requiring disambiguation.

Even if the ray cursor does provide better performance within volumetric displays, it has an inherent problem associated with its use which requires exploration. In dense target environments, the ray may intersect multiple objects, and so the actual target of interest is ambiguous (Figure 6-1). While techniques for disambiguation have been proposed [Hinckley et al. 1994, Liang and Green 1994, Olwal and Feiner 2003, Steed 2006, Wyss et al. 2006], few have been implemented, and none appear to have been systematically evaluated. Furthermore, the proposed solutions have drawbacks of their own, such as introducing extra buttons or modes for cycling mechanisms [Hinckley et al. 1994], or having the system use heuristics to make predictions about the intended target [Liang and Green 1994, Steed 2006], which may not be accurate, or fail under certain environment layouts.

In this chapter, we design and evaluate selection techniques appropriate for volumetric displays. We first implement and evaluate the 3D point cursor and ray cursor in a single target volumetric display environment. Consistent with the VR literature the ray cursor was found to have faster selection times than the point cursor. We then present four different design enhancements to the ray cursor which allow users to disambiguate multiple objects, namely, the depth ray, lock ray, flower ray, and smart ray. We evaluate these new techniques in a second experiment, under a dense target environment. Except for the predictive smart ray technique, our new techniques were successful, each with their own beneficial properties. In particular, our depth ray technique significantly reduced movement times, error rates, and input device footprints. We discuss the implications of our work to user interface design, and conclude with some remarks about future lines of work.

6.2 Related Work

In 3D virtual environments, selection is categorized as one of the four basic interactions (along with navigation, manipulation, and system control) [Bowman and Hodges 1999, Mine 1995a]. A user must be able to specify an object in the virtual world, so they can then manipulate or interact with it. In Section 2.6.2 we provided a thorough discussion of the research relevant to 3D selection techniques. Here we summarize those results which are most relevant to the work presented in this chapter, and discuss how our work will extend the prior art.

Liang and Green [1994] implemented a ray firing selection mechanism that they called "laser gun" selection (Figure 2-10b). A ray is emitted from the user's hand, so the user has control over the start point and orientation of the ray, much like a physical laser pointer. With this technique it was found to be difficult to select small and distant objects due to the required angular accuracy. To alleviate this problem, they created a mechanism called "spotlight selection" where instead of emitting a ray the user emits a conic selection area, with its apex at the user's hand. Other related forms of ray cursors have also been developed, such as aperture based selection [Forsberg et al. 1996] and 2D image plane selection [Pierce et al. 1997].

Generally with the ray cursor techniques, only the first intersected object will be selected, even though the ray can intersect multiple objects simultaneously. Under this implementation, it could be very difficult or even impossible to select objects that are further away, depending on the density of the target environment. While no evaluation has been conducted to examine this, the following techniques have been proposed.

Liang and Green [1994] developed a metric for the spotlight selection to decide which object would be selected if multiple targets were within the cone, based on the distance between the target to the apex and central axis of the cone. While this metric may work in some situations, it would fail for objects further away from the user, if there were a number of closer targets along the same line.

An interesting extension to spotlight selection is Shadow Cone Selection [Steed and Parker 2004], which selects targets by sweeping out an area with a cone selection cursor. While useful for group selections, the shadow cone does not provide a disambiguation mechanism, as all targets which are intersected by the sweep will be selected.

Hinckley et al. [1994] suggest that the ray casting technique could be augmented with a mechanism for cycling through the set of all ray-object intersection points. While this would allow the user to specify the correct target regardless of the density of the environment, it would require extra buttons, it could be cumbersome if there were a large number of targets to cycle through, and it is not clear how the cycling modality would be incorporated into the selection modality

Olwal and Feiner [2003] describe the flexible pointer, a ray cursor technique which allows users to point to objects which are fully or partially occluded. Users can bend the ray cursor so that it points to their target of interest, without passing through distracter targets. However the technique requires two six degree-of-freedom devices to control the cursor, and also requires the user to specify the 3D location of the desired target. Another technique requiring two input devices is iSith [Wyss et al. 2006], where two rays are simultaneously controlled, with their intersection being used to define a target location.

Steed [2006] suggests several new methods for disambiguating multiple targets, such as improving the metrics proposed by Liang and Green [1994] for spotlight selection, or gesturing such that the target of interest remains in the selection area over a period of time. While interesting ideas, it is unclear if these techniques were implemented or evaluated.

A more direct method of interaction, in which disambiguation is not an issue, is to use a 3D point cursor which specifies X, Y and Z coordinates for 3D selections [Hinckley et al. 1994, Mine 1995a, Poupyrev et al. 1996]. This is the type of cursor used in our studies in Chapter 5, and is illustrated in Figure 5-3. Mine [1995a] states that in local interactions, a direct mapping from the user's hand to a 3D "virtual cursor or drone" could be used to select an object. The benefit of this method is that it is completely unaffected by the target density of the environment. The problem, however, is that the selections are constrained by three dimensions, resulting in longer selection times. This has been confirmed in a previous study [Bowman et al. 1999], as well as in our work in Chapter 5. Instead of a 3D point cursor, Zhai et al. [1994] developed the silk cursor, which is a 3D volume cursor (Figure 2-12). While using a volume cursor could reduce target acquisition times, it once again produces a difficulty when interacting in dense target environments, as multiple targets may fall within the bounds of the cursor's volume. Although never implemented, a number of the above techniques were discussed for use within volumetric displays in an exploratory paper with wizard-of-oz prototypes [Balakrishnan et al. 2001].

There has also been recent work in disambiguating multiple targets in dense two-dimensional target environments. The bubble cursor [Grossman and Balakrishnan 2005a] is an area cursor that dynamically changes its size and shape to always capture only the closest target. The splatter technique [Ramos et al. 2006] allows users to spread out overlapping 2D objects, to reduce occlusions. Both of these techniques provide inspiration for the dense environment 3D selection techniques which we propose and evaluate.

6.3 Experiment 1: Sparse Environment

The main goal of this study is to obtain data on known selection techniques for volumetric displays, in a simplified and controlled single-target environment. We will compare the two most popular candidates from the VR literature, the point cursor and the ray cursor. The data which we obtain from this experiment on these two techniques will be used to guide the design of our new techniques, suitable for more realistic, dense target environments.

Although the ray cursor has previously been shown to outperform direct pointing on other platforms [Bowman et al. 1999], it is not clear if this will hold within the volumetric display. Unlike most immersive VR environments, in a volumetric display all targets are within arm's reach, and so the required distance to travel to any target will be minimal. Our work in Chapter 5 showed that the time taken to select an object using the 3D point cursor follows Fitts' Law, and will thus be a function of this travel distance. It is, therefore, of interest to determine if this minimized distance property of volumetric displays is significant enough for the performance of the point cursor to surpass the performance of the ray cursor.

6.3.1 Apparatus

As with the studies described in the previous chapters, we used a 3D volumetric display developed by Actuality Systems (www.actuality-systems.com). The display is fully described in Section 1.4. The experiment was run on a 2 GHz Pentium4 computer. For input, an Ascension Flock-of-Birds six degree-of-freedom tracker equipped with a single button was used, which had a positional accuracy of less than 2mm, and operated at 100Hz.

6.3.2 Participants

Six male and six female unpaid volunteers, ranging in age from 23 to 35, served as participants in this experiment. Participants were computer science graduate students. Participants were screened through the Stereo Optical RADNDOT stereopsis test. One of the twelve participants was left handed and the rest were right handed. Participants controlled the tracking device with their dominant hand. Two of the participants had more than 5 minutes experience viewing the volumetric display, and only one of the participants had some experience interacting with it. Six of the twelve participants had experience with 3D tracking devices, but none of them reported this experience being more than "little".

6.3.3 Procedure

A 3D static target acquisition task was used. Targets were rendered as yellow wireframe spheres. A start target would randomly appear in one of the eight octants of the display. Once this target was selected the trial began, and a goal target would appear at a random position in one of three possible octants which were not adjacent to the starting octant. The distance between the start and goal targets was always set to 5 inches. Subjects were told to complete the trials as quickly as possible, while minimizing errors. The radius of the start target was always 0.45 inches, and the end target took on a radius of either 0.3 or 0.6 inches. For control purposes, users were centered in front of the display and were told not to move their feet during the trials.

The 3D point cursor was displayed as a crosshair with short line segments along the three main axes. The handheld tracker controlled the 3D cursor with an absolute one-to-one mapping and a control-display ratio of one. To select a target with this cursor, the center of the crosshair had to be positioned inside of it (Figure 6-2a, b). The ray cursor was displayed as a single line segment, originating at the surface of the display. The input device controlled both the orientation and origin position of the ray, with a direct one-to-one mapping. To select an object with the ray cursor the ray had to intersect the target (Figure 6-2c, d). For both cursors, selection was initiated by clicking the button.

We also included two common forms of visual feedback, to ensure that the presence or absence of such feedback would not affect the relative performance of the techniques.

The feedback which we included were highlighting [Mine 1995a], where a target color would change to red when it could be selected, and shadowing [Wanger 1992], where a 2D projection of both the target and cursor were displayed on a ground plane.

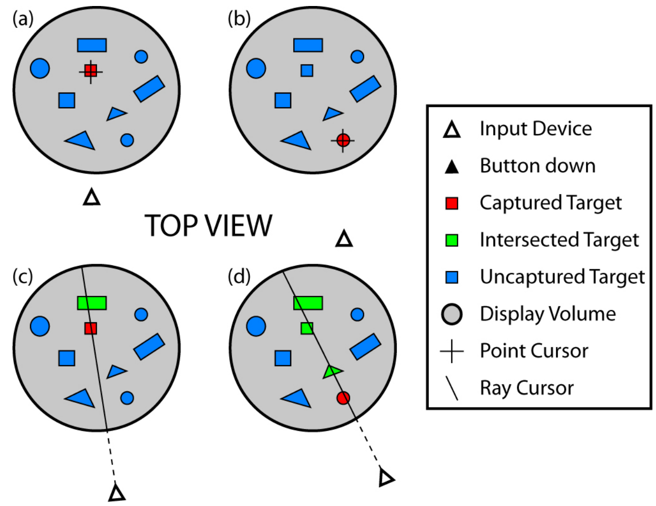

Figure 6-2. Experiment 1 selection techniques. (a-b) The point cursor is controlled by the position of the input device. A target can be selected by positioning the crosshair inside of it. (c-d) The ray cursor is controlled with the position and orientation of the input device. The ray cursor will select the first target which it intersects. This legend will be used for the remainder of the chapter.

6.3.4 Design

A balanced within subjects design was used. The independent variables of interest were cursor type *CT* (point cursor, ray cursor), visual feedback *FB* (none, highlighting, shadowing), and goal target size *SIZE* (0.3, 0.6). The 12 combinations of *CT*, *FB*, and *SIZE* were fully crossed with 8 possible start target positions, each with 3 possible goal target positions, resulting in a total of 288 combinations. Participants were divided into two groups of six. Order of presentation of cursor type was counterbalanced across the groups, with all trials for the first cursor type performed in a single session on one day, and all trials for the second cursor type in a second session on another day. Each session lasted approximately one hour. Within each group, participants were randomly assigned one of the six unique orderings of the three *FB* conditions.

Before each session there was a 2 minute demonstration and warm up. Participants completed all trials for each of the three *FB* values in a session, with trials for each value of *FB* being repeated in four blocks, for a total of 12 blocks per session. In each block, the start and goal target positions, along with the goal target size were presented in random order. A short break was taken between each block.

6.3.5 Results

Task performance was measured by movement time, defined as the time between successful selections of the start and end targets. In our analysis of movement time we removed outliers more than 3 standard deviations from the group mean (1.8% of data) and trials in which errors occurred (11.3% of data). The error rates were not significantly affected by the cursor type or visual feedback.

Analysis of variance indicated that *CT* ($F_{1,11} = 2180$), *FB* ($F_{2,22} = 105.4$), *SIZE* ($F_{1,11} = 3112$), all significantly affected movement time at the $p < .0001$ level. Pair-wise means comparison using Tukey's post-hoc test showed that the ray cursor was significantly faster, with overall movement times of 1.27s for the ray cursor, and 1.62s for the point cursor. There was also a significant *CT* x *SIZE* interaction ($F_{1,11} = 54.84$, $p < .0001$), as illustrated in Figure 6-3. Post hoc analysis shows that ray cursor is significantly faster for both sizes at the $p < .0001$ level. It can also be seen that movement times for point cursor increase to a greater extent than for the ray cursor, when acquiring the smaller target. This interaction is an interesting effect, as it indicates that the ray cursor is less affected by the size of its goal target.

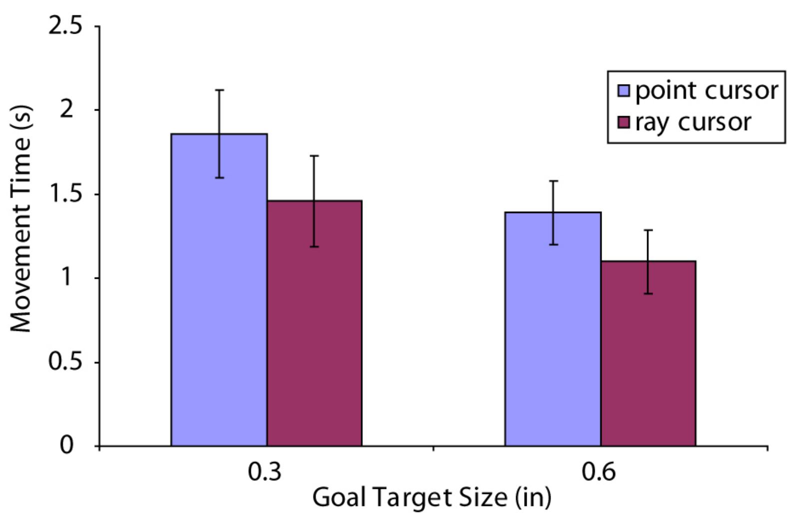

Figure 6-3. Movement times by target size.

Although there was a significant effect for *FB* on movement time, the feedback did not improve movement times. The times were 1.42s for highlighting and 1.43s for none, which were not significantly different, and 1.53 for shadow, which was significantly higher than the other two feedback conditions ($p < .01$). It is interesting that the highlighting feedback did not improve movement times, showing that the users could suitably perceive when the ray intersected the goal target, and when the point cursor was inside the goal target. It should also be noted that this was not due to a speed accuracy tradeoff, as *FB* did not have a significant effect on error rate. The increased movement times in the shadowing condition were likely due to divided attention.

6.3.6 Summary

Consistent with the previous VR literature, we have found that the ray cursor results in significantly faster selection times than the point cursor, even with limited travel distances within the volumetric display. Moreover the ray cursor was less affected by a reduction in target size, likely because a reduced target size means only two dimensions of motor space are reduced, while for the point cursor, three dimensions of motor space are reduced. As for the visual feedback, neither form reduced movement times, or affected the cursors differently.

While the results of Experiment 1 are useful, we are still left with the goal of finding a 3D selection technique which can be effective in both sparse *and* dense target environments. The data clearly shows that the ray cursor is better for sparse environments. As such, we are further motivated to explore enhancements to the ray cursor which provide disambiguation mechanisms for dense target environments. We now provide a detailed description of the new techniques which we have designed.

6.4 Ray Cursor Disambiguation Techniques

As previously discussed, a limitation with the ray cursor is that if it simply selects the first target which is intersected, then it could be difficult or even impossible to select occluded targets in a dense environment. This problem is especially important for volumetric displays, as the selection of occluded objects may be more common for the following two reasons.

Firstly, because the display is so close to the user, the user's viewpoint vector will be drastically different from the ray cursor vector, which is emitted from the hand. So even though targets may not be occluded from the user's point of view, they may be occluded relative to the vector of the ray cursor. Secondly, even if targets are occluded from the user's viewpoint, the user will still be able to see them, as all imagery is semi-transparent in the current generation of volumetric displays.

In the following sections, we present four new selection techniques, based on the ray cursor, which allow for the disambiguation of multiple intersected targets. The depth ray, lock ray and flower ray require explicit disambiguation, while with the smart ray the disambiguation is implicitly performed by the system. For the explicit disambiguation techniques we explore both concurrent (depth ray) and sequential (lock ray and flower ray) selection and disambiguation phases.

6.4.1 Depth Ray

The depth ray augments the ray cursor with a depth marker, visualized as a small sphere, existing along the length of the ray (Figure 6-4a). Along with the standard control of the ray cursor, the position of the depth marker can also be controlled dynamically. The distance between the hand and the surface of the volumetric display is mapped to the position of the depth marker, using an absolute mapping. Moving the hand forwards and backwards will move the depth marker in the same manner (Figure 6-4b, c).

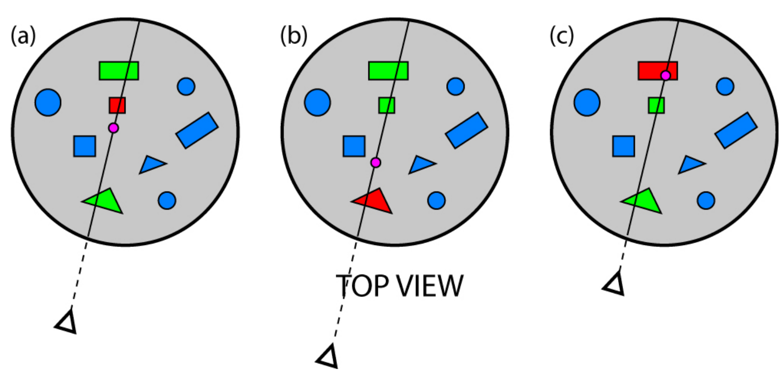

Figure 6-4. The depth ray. (a) The depth ray selects the intersected target which is closest to the pink depth marker. (b) Moving the input device backwards selects a closer object. (c) Moving the input device forwards selects the further target. Note that the depth marker can intersect a target to select it, as in (c), but isn't required to, as in (a) and (b).

With the depth ray, all targets which are intersected by the ray are highlighted green. Of these intersected targets, the one which is closest to the depth marker (illustrated as a pink circle in Figure 6-4) is highlighted red, indicating that it will be selected with a button click. Note that instead of discretely cycling from one target to the next, as suggested by Hinckley et al. [1994], we chose to continuously move the depth marker along the length of the ray and select the closest target. This design was inspired by the bubble cursor [Grossman and Balakrishnan 2005a], a 2D selection technique which moves around the screen continuously, and selects the closest target. This technique was shown to outperform the object pointing technique [Guiard et al. 2004], which jumps from one target to the next.

6.4.2 Lock Ray

The depth ray allows users to control the position and orientation of the ray, while simultaneously disambiguating between multiple targets. While this may allow for fast selections, the two phases could potentially interfere with one another. Adjusting the ray position could cause the depth marker to move and vice-versa. As a solution to this, we developed the lock ray, a similar technique, but the selection and disambiguation phases are carried out sequentially, in a two-step process.

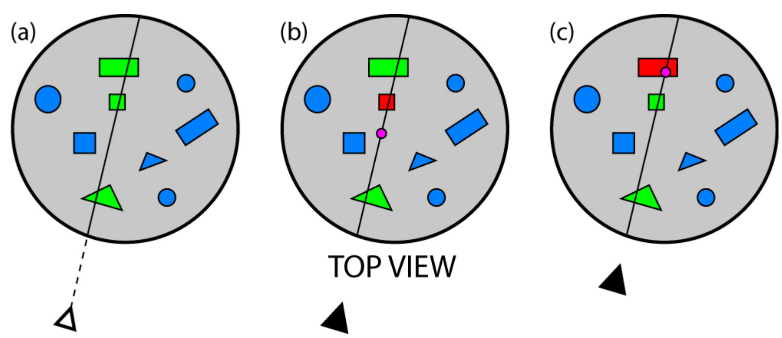

Figure 6-5. The lock ray. (a) All intersected targets are highlighted. (b) Holding the button down locks the ray and displays the depth marker at its center. (c) The depth marker is controlled with the input device, selecting the closest intersected target.

With the lock ray, all intersected targets are also highlighted green; however no depth marker is visualized (Figure 6-5a). To specify the target, the user clicks and holds the button down. At this point, the position of the ray is locked, and only then does the depth marker appear (Figure 6-5b). The user adjusts the depth marker in a similar manner to the depth ray, and the intersected target which is closest to the depth marker is highlighted red indicating that it can be selected by releasing the button (Figure 6-5c). Keeping the button down during the disambiguation phase provides a kinesthetically held mode, avoiding confusion between selection and disambiguation phases [Sellen et al. 1992].

The initial position of the depth marker is always the center of the ray, so users will know which way they need to move it to acquire their goal target, even before it is visualized. In the event that the user misses the goal target when the ray is locked, the user's hand can be moved in a direction perpendicular to the ray to cancel the selection.

6.4.3 Flower Ray

The flower ray is another two-step selection technique, similar to the lock ray. The selection phases of the techniques are the same (Figure 6-6a). However, with the flower ray, when the user clicks and holds the button, all intersected targets animate towards the user's viewpoint, and flower out into a marking menu [Kurtenbach and Buxton 1993] (Figure 6-6b). The rationale behind this design is that a marking menu selection should be faster than the disambiguation phase of the lock ray, which is much like selecting an item from a linear menu. This technique is a 3D extension to the splatter technique [Ramos et al. 2006], which spreads out items which are occluded in a 2D layout when the user clicks down.

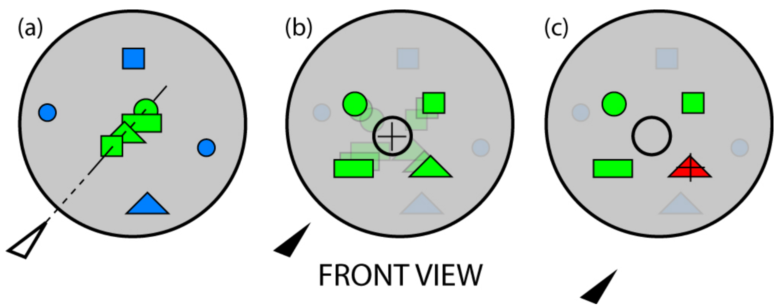

Figure 6-6. The flower ray. (a) All intersected targets are highlighted. (b) Holding the button down causes all intersected targets to flower out into a marking menu. (c) The input device is used to select the desired target from the marking menu.

When the marking menu appears, a 2D cursor, controlled by the input device, is drawn in the center of the menu. The cursor needs to travel a minimum distance, visualized by a circle, to select any of the targets. Once leaving the bounds of the circle, the target closest to the 2D cursor will be highlighted red, indicating that it can be selected by letting go of the button (Figure 6-6c). As with the lock ray, a selection can be cancelled if the intended target was not selected and does not appear in the marking menu. To do so, the button is released while the cursor is still inside the bounds of the circle.

While the marking menu will potentially make the flower ray faster than the lock ray, a possible drawback is that users will need to follow the animation and find their intended object in the marking menu. This is not an issue with the lock ray since the disambiguation phase is completed in place. To minimize this effect, we arrange the targets in a clockwise fashion about the marking menu, in order of their depth from the user. The closest target is displayed at the top right, the furthest target is displayed at the top left, and remaining targets are distributed evenly.

6.4.4 Smart Ray

So far, the new techniques which we have described all require an explicit disambiguation phase carried out by the user. We felt that giving user explicit control would be the correct approach, as research in 2D selection interfaces have shown that predictive techniques can be detrimental to performance [Grossman and Balakrishnan 2005a]. However, for the sake of comparison we decided to also include a predictive technique in which the disambiguation phase is performed implicitly by the system. Previously implemented predictive techniques rely on a metric based on the current ray position within the target layout [Liang and Green 1994, Steed 2006]. However such an approach does not guarantee that every target can be selected. The target could exist in an environment such that no matter how the ray intersects the intended target, another target is intersected and selected by the algorithm. As such, we feel it is necessary for the prediction algorithm to be based on the history of the ray cursor's movements.

The design of the smart ray is based on the idea that the intersection of two rays could define a point in 3D space. Instead of taking the intersection of two simultaneously defined rays, which would require a second input device [Wyss et al. 2006], the smart ray takes the intersection of a single ray over a length of time. This technique was recently proposed but not implemented or evaluated [Steed 2006].

In our implementation, we use an algorithm based on target weights to determine which target should be selected when multiple targets are intersected. Target weights are continuously updated based on their proximity to the ray cursor, and are visualized with small spheres at the center of the target (Figure 6-7a). The closer the ray comes to the center of the target, the larger the weight increase will be.

As with the previous techniques, all intersected targets are highlighted green. The intersected target with the highest weight is highlighted red, indicating that it can be selected by clicking the button. By using this algorithm, when the ray intersects multiple targets, the user can reposition the ray so that its new position still intersects the intended target (Figure 6-7b). Even if multiple targets are selected by the new ray position, the intended target will have the highest weight, as its weight has been continuously increasing (Figure 6-7c).

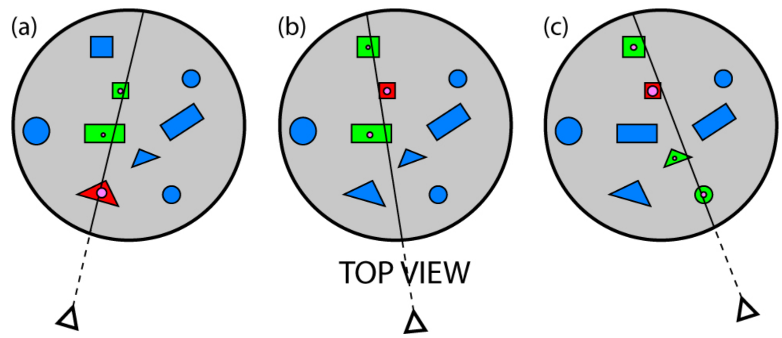

Figure 6-7. Using the smart ray to select the small square. (a) Target weights are based on the distance from the ray to the target, visualized as spheres in the center of each intersected target. The target with the highest weight can be selected. (b-c) The ray can be repositioned to select an occluded target, by continually increasing its weight.

This technique is similar to the shadow cone [Steed and Parker 2004]. However the shadow cone requires that targets remain intersected during the entire selection. In our initial pilot studies, it was clear that this constraint was much too strong, as it was difficult for users to reposition the ray in a manner that their goal target was intersected the entire time. The smart ray relaxes this constraint. Weights will gradually decrease when the ray is moved away from a target, but as long as the target is reacquired within a reasonable amount of time, it will have the highest weight and can be selected.

6.5 Experiment 2: Dense Environment

In Experiment 1, we found that the ray cursor is a faster selection technique in comparison to the point cursor in a single target environment. However the ray cursor, in its naïve implementation, is not an appropriate technique for volumetric displays, in a realistic usage scenario, due to the problem of multiple target ambiguity. Motivated by this difficulty, we have presented the design of four new selection techniques, all based on the ray cursor, which provide mechanisms for disambiguating multiple targets.

All of these new techniques have both potential benefits and drawbacks. The depth ray integrates the selection and disambiguation phases, which could minimize times, but could also cause interference between phases. The lock ray explicitly separates the phases, but the disambiguation is accomplished with a linear menu selection. The flower ray provides a marking menu for disambiguation, which should be faster, but users need to follow an animation and find their intended target in the marking menu. Finally, the smart ray provides an implicit and possibly more fluid disambiguation mechanism, but as with any predictive user interface, it could cause frustration if the system misinterprets the user's intent.

In an effort to evaluate the relative effect of these potential benefits and drawbacks, we now present a second experiment, evaluating our four new techniques. The experiment will be conducted in a dense target environment, designed such that it would be virtually impossible to select the target without a disambiguation mechanism. As a result, we omit the naïve implementation of the ray cursor. Instead, we use the point cursor as the baseline for comparison with our new techniques, as its performance should remain unaffected by the density of the environment. This will allow us to identify which, if any, of our new techniques are still faster than using the point cursor, even with the addition of the disambiguation mechanisms. Such techniques, if they perform well, would be appropriate for use within volumetric displays, as they would provide fast selection for both sparse and dense target environments.

6.5.1 Apparatus

Experiment 2 was run on the same volumetric display and computer as in Experiment 1. The tracking technology differed as the experiment was carried out at a later time. The input device was a wireless single-button presentation mouse. Three passive-reflective markers were placed on the device, which were tracked in 3D by a Vicon motion tracking system (www.vicon.com). This allowed us to track both the 3D location and orientation of the input device. The markers were tracked at 120Hz with sub-millimeter precision. This input device is illustrated in Figure 6-1.

6.5.2 Participants

Eight male and two female new unpaid volunteers, ranging in age from 20 to 25, served as participants in this experiment. Participants were all computer science undergraduate students. Participants were screened through the Stereo Optical RADNDOT stereopsis test. All were right handed and controlled the input device with their right hand. None of the participants had previously viewed a 3D volumetric display, however all participants had some experience with either 3D graphics or 3D games. Two of the participants had experience using 3D input devices.

6.5.3 Procedure

As with Experiment 1, a 3D static target acquisition task was used. To begin a trial, users selected a sphere displayed at the front of the display. After clicking this target, the experiment environment was displayed, consisting of a 3x3x3 array of distracter targets, and a single goal target (Figure 6-8). The goal target was rendered as a yellow wireframe sphere, and the distracter targets were rendered as blue wireframe tetrahedrons. Since we were mainly interested in the disambiguation component of the techniques, we kept the goal target size constant, with a radius of 0.3 inches. Distracter targets were larger, ensuring that when using the ray cursor techniques to select the goal target, distracter targets would have to be intersected. As in Experiment 1, users were centered in front of the display and were told not to move their feet during the trials.

The goal target was positioned behind one of the distracter targets, either in the left or right row. This resulted in 18 possible target locations (Figure 6-8). Participants had to successfully select the goal target to complete a trial. Selection errors occurred if the user selected either the wrong target or no target at all.

For consistency across all cursor types, targets were not considered selected by the point and ray cursors until the button was released. A conic selection area was used for the ray cursors, with a 2-degree angle at the apex of the cone. This increased the number of targets which would have to be disambiguated, as more distracter targets would be intersected. However, our initial observations of informal usage showed that the benefit of the conic selection outweighed the cost of having to disambiguate between a few more targets. Although a conic selection area was used, the cursor was still rendered as a single ray. Target highlighting occurred for each technique, as described by the individual techniques is Section 6.4.

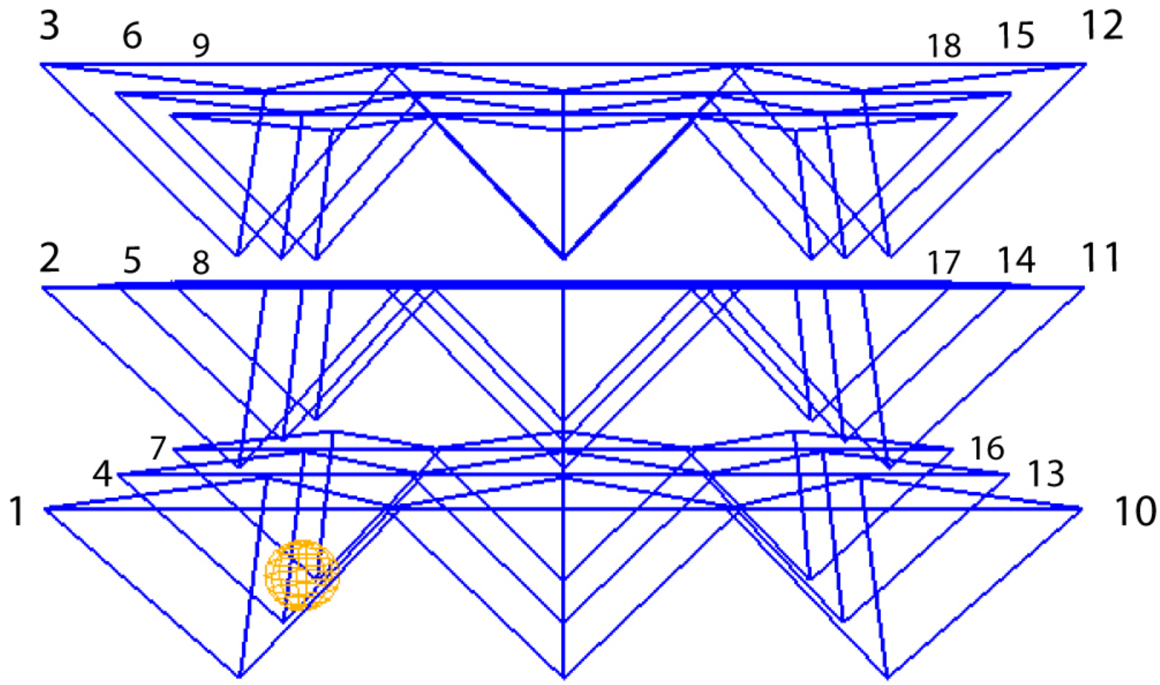

Figure 6-8. Target environment for Experiment 2, consisting of a sphere goal target and a 3x3x3 array of tetrahedron distracter targets. Target location numbers correspond to which of the 18 distracter targets the goal target is behind.

6.5.4 Design

A repeated measures within-participant design was used. The independent variables were the cursor type *CT* (point cursor, depth ray, lock ray, flower ray, smart ray) and target location *LOC* (1-18). The experiment lasted approximately 90 minutes, and was divided into 5 sessions, with short breaks in between sessions. Each session consisted of all trials for one of the five values of *CT*. Sessions were broken up into 3 blocks of 54 trials, with the 18 target locations appearing 3 times each in random order. This design resulted in 810 trials per participant.

To familiarize participants with the task and selection techniques, eight warm-up trials were performed before each session began. Presentation orders of the selection techniques were counterbalanced using a 10x5 balanced Latin square design. Participants were randomly assigned one of the 10 orderings.

6.5.5 Results

The main dependent measures for the task were trial completion time, error rate, and input device footprint. Trial completion time can be further analyzed into the selection phase time and disambiguation phase time.

6.5.5.1 Trial Completion Time

In our analysis of trial completion time, we discarded trials in which errors occurred (13.3% of data), and removed outliers that were more than 3 standard deviations from the group mean (1.6% of data).

Repeated measures of analysis of variance showed main effects for *CT* ($F_{4, 36} = 188$), *LOC* ($F_{17, 153} = 16.5$), and the *CT* x *LOC* interaction ($F_{68, 612} = 9.23$) (all $p < .0001$). Average trial completions times were 3.51s for the smart ray, 2.69s for the lock ray, 2.54s for the point cursor, 2.46s for the flower ray, and 2.05s for the depth ray (Figure 6-9). Pair-wise means comparison using Tukey's post-hoc test showed that the point cursor was not significantly different from lock ray or flower ray, but all other pairs were significantly different ($p < .001$).

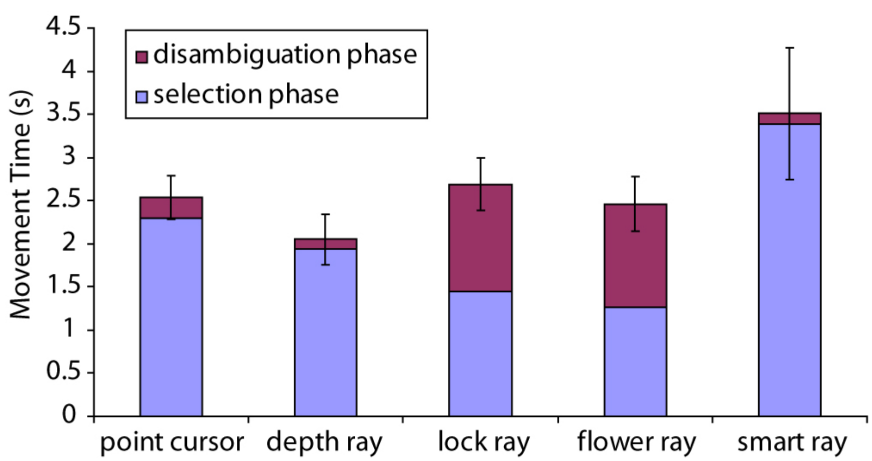

Figure 6-9. Movement times for each cursor, by the selection and disambiguation phases. Error bars illustrate 1 standard deviation of the total trial time.

Figure 6-10 shows the movement times for each cursor by the goal target location. The most prominent effect seen here is that movement times for the smart ray were similar to other techniques for targets on the left side of the display, but much worse when targets were on the right side of the display. This may seem strange since the environment was completely symmetrical. However, because users were right handed, the ray was also coming from the right side. It is clear that due to the arrangement of targets, under this condition, the predictive algorithm broke down.

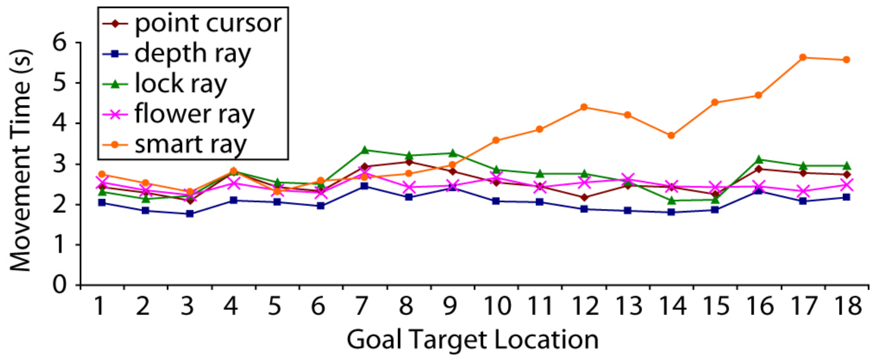

Figure 6-10. Movement times by target location.

6.5.5.2 Trial Phase Times

Some interesting effects are seen when we break the data up by the two phases. We define the selection phase as the time until the user clicks the button down. The disambiguation phase is the subsequent time until the button is released. For the point cursor, depth ray and smart ray, the disambiguation phase times will be minimal, only consisting of the time taken to click the button.

Figure 6-9 breaks the total movement times down by the two phases. Selection phase times were significantly affected by *CT* ($F_{4, 36} = 433$, $p < .0001$). As expected, the times were slower for the three techniques for which the disambiguation and selection are done concurrently ($p < .0001$). Of these three techniques, the depth ray was significantly fastest, followed by the point cursor and then the smart ray ($p < .0001$). The flower ray was slightly faster than the lock ray ($p < .01$), which is surprising, since the techniques are exactly the same during the selection phase. With the lock ray, we suspect that users were likely planning their disambiguation movements before completing the selection phase, causing the increase in time. In comparison to the flower ray, the selection phase of the depth ray was 0.69s slower, which is the added cost of integrating the disambiguation and selection phases for that technique.

When looking at the disambiguation phases, the lock ray is only slightly slower than the flower ray ($p < .05$). This shows that with the flower ray, the animation time, and the time to find the target in the marking menu, negates the advantage of using a marking menu.

However the advantage with the flower ray is that disambiguation times are more stable, regardless of the target location. Indeed there is a significant *CT* x *LOC* interaction for these techniques ($F_{17, 153} = 39.5$ $p < .0001$) (Figure 6-11). It can be seen that the location has much more effect on the lock ray, and less effect on the flower ray. The disambiguation times for these techniques, which are the added costs of a sequential disambiguation phase, are 1.20s for the flower ray and 1.25s for the lock ray. This is much higher than the added cost of 0.69s for the concurrent disambiguation phase of the depth ray, which is why the depth ray was fastest overall.

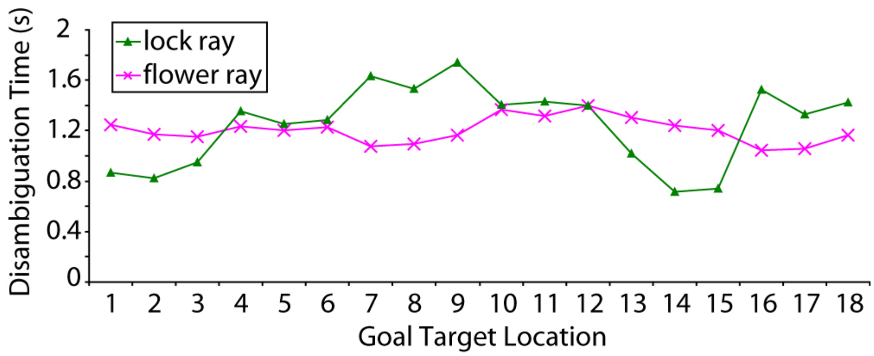

Figure 6-11. Disambiguation times for the flower ray and lock ray.

6.5.5.3 Learning

A slight learning effect was seen, with the block number significantly affecting movement times ($F_{2, 18} = 10.6$, $p < .001$). Block 1 had the slowest times, averaging 2.78s. Blocks 2 and 3 were significantly faster than Block 1, but not from each other, with average times of 2.58s and 2.61s respectively. There was no interaction between the block number and cursor type. This shows that our new techniques were just as easy to learn as the 3D point cursor.

6.5.5.4 Input Device Footprint

Another variable which we measured was the input device footprint. We measured the length of the total path which the device took to complete the trial. The cursor type had a significant effect on the input device footprint ($F_{4, 36} = 56.5$, $p < .0001$). Figure 6-12 illustrates the effect. Pair-wise means comparison using Tukey's post-hoc test showed that the lock ray, flower ray, and depth ray all had significantly lower footprints than the point cursor ($p < .0001$), while the footprint of the smart ray was significantly higher (p <

.0005). The reduction of footprint is especially important since a handheld six degree-of-freedom device is being used, which can lead to fatigue with extended use [Ware and Slipp 1991].

6.5.5.5 Error Rate

With respect to errors, all of our new techniques performed better than the point cursor. The point cursor had a particularly high error rate of 20.7%. The error rates for the ray cursor techniques were all significantly lower; 13.3% for the depth ray, 11.1% for the lock ray, 10.9% for the flower ray, and 10.4% for the smart ray (all $p < .05$).

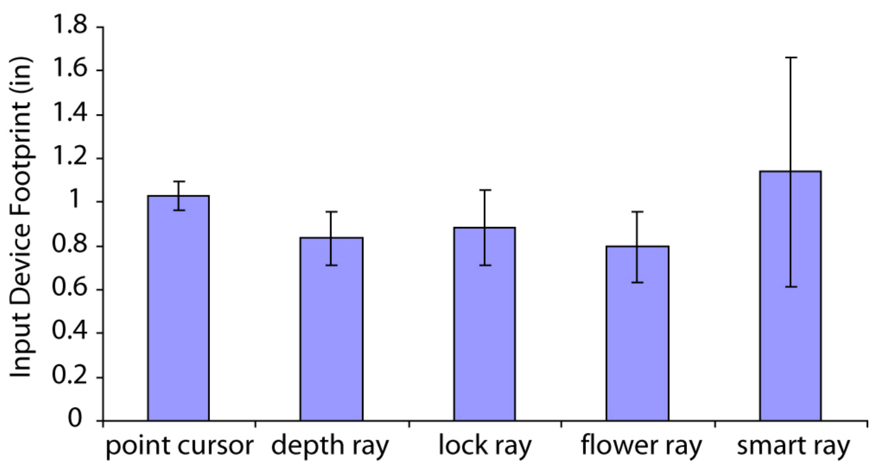

Figure 6-12. Input device footprints.

6.6 Discussion

The results of the studies conducted in this chapter indicate that ray cursors will be faster than 3D point cursors in volumetric displays, and that the depth ray is an effective implementation of the ray cursor for providing disambiguating of targets. The results of the first study are consistent with prior research in virtual reality and 3D interaction. Mainly, that a technique which reduces the dimensional constraints of a selection task will be faster [Bowman et al. 1999, Ware and Lowther 1997]. However, because of the reduced distances in volumetric displays, it was important to ensure that this result would be maintained.

The work presented in this chapter builds upon the formal modeling of 3D pointing which was done in Chapter 5. However, we are unaware of similar models for pointing with the ray cursor. As such, while we do have explanations for why the ray cursor was faster than the point cursor, and why the depth ray outperformed the other techniques, we do not have theoretical models to refer to in this explanation. Developing such models could be useful for determining which techniques will be superior under varying conditions.

6.7 Implications for User Interface Design

The results of our study have clear implications for the future design of volumetric display user interfaces. A ray cursor metaphor should be used, as it will improve movement times, lower error rates, and reduce the input device footprints, for the common 3D selection task. As for the disambiguation technique, the decision should be based on a couple of considerations.

Most importantly, the input device which will be used should be taken into account. This is because the depth ray requires an input device which has enough degrees of freedom to specify the ray *and* specify the depth, since these phases are completed simultaneously. In contrast, the flower ray and lock ray only require enough degrees of freedom to specify the ray, since the disambiguation phases are completed independently. We will discuss input devices for which these techniques may be more appropriate in the next section.

Another implication of our results is that predictive techniques, such as the smart ray, should probably be avoided. The smart ray performed poorly based on all measurements. Although in theory the technique should have worked well, the results showed that its performance was highly affected by the location of the goal target within the environment. The poor performance was due to two factors. First, it was difficult for users to keep the ray close to the target while moving from one position to another. Second, users preferred to minimize their hand movements, so the change in ray angles was not drastic enough for the technique to work properly.

There are a number of parameters involved with the smart ray algorithm which were chosen in an effort to maximize its usability. Further experimentation could be conducted to optimize these parameters. However, based on our results, it seems unlikely that this would reduce selection times to the extent which our other more successful techniques have. The results for the smart ray were not surprising, as it has been similarly found that predictive selection techniques in 2D can provide poor performance [Grossman and Balakrishnan 2005a].

While our study was focused on selection techniques for volumetric displays, the implications extend to other technologies, such as VR environments. If the depth ray or lock ray were used, then acceleration mappings for the depth marker could be required [Poupyrev et al. 1996], since intersected objects could drastically range in distance. The flower ray would not be affected by this, possibly making it the most appropriate technique for large VR environments.

6.8 Future Work

We have a provided a number of new selection techniques for volumetric displays, which work in both sparse and dense target environments. While three of these techniques performed well, there are areas to explore in the future. In particular, we now discuss alternative input devices which could be used for selection, and alternative techniques which we did not consider.

6.8.1 Alternative Input Devices

Our study focused on using a six degree-of-freedom input device for selection. This was required to support both the 3D point cursor, which utilizes positional information, and the ray cursors, which utilize orientation information. If such a device was going to be used, our results suggest that the depth ray would be the most appropriate technique. However the physical form of the device remains to be investigated.

The device could take on the pen shape of a laser pointer, to reinforce the metaphor of emitting a virtual light ray. Alternatively, the system could track the user's index or "pointing" finger, which could specify the location, orientation and depth of the ray. This would eliminate the need for an auxiliary input device, but would also eliminate the buttons necessary to confirm a selection. One possible solution would be to use freehand gestures to simulate button clicks. We will explore this idea in the next chapter.

Another possible option, which diverges from six degree-of-freedom input, is to use the surface of the display as the input device. Layering the enclosure with a touch sensitive surface would allow the user to manipulate the selection cursor by either directly touching the display surface or using a stylus. If the surface only sensed the contact position, than the input would only have two degrees of freedom. With such input, the position of the ray could be defined by the contact point, and the ray could be made to always go through the center of the display volume. This would allow the user to intersect any location in the volume. With such a setup, the depth ray would be inappropriate, as there would be no obvious way to control the position of the depth marker. However, the flower ray would work well, as the marking menu stroke could be made along the surface of the display once the selection phase was completed. Similarly, the lock ray could be used, with the depth marker being specified with linear scrubbing on the display surface. This technique may be problematic, as our results show that users prefer to minimize the movements of their hands. If the ray always went through the center of the display, large movements may be required to intersect specific locations. However, if the surface also sensed tilt, then the user could also have control over the orientation of the ray. If the surface were also pressure sensitive, then there would be five degrees of freedom, enough to possibly use the depth ray, controlling the position of the depth marker with pressure.

Another input device for which the lock ray or flower ray would be more appropriate is a mouse with extra degrees of freedom, such as the four degree-of-freedom Rockin'Mouse [Balakrishnan et al. 1997]. The Rockin'Mouse could control the position of the ray through positional movement and the orientation of the ray through tilt. Once completing the selection, subsequent positional movements could be used to specify the depth of the lock ray or make the marking menu selection for the flower ray. This technique has the advantage of leveraging the many beneficial properties of the traditional mouse. Most relevant to our work, arm fatigue would be reduced, in comparison to controlling a device in free space, or interacting directly with the display surface.

One last input configuration which may also be interesting to explore is multiple pointing devices, as an alternative to the smart ray. The user could control two rays, one by each hand, and the midpoint of the closest line which connects the two rays could be used as a selection point. One potential advantage of this technique is that it could be used to increase precision of the input. This effect has been demonstrated by multifinger selection techniques in 2D environments [Benko et al. 2006].While this option was considered for our study, we chose not to include it for the pragmatic reason that we believe that two hands should not be required for a task as common and simple as target selection. While adding a second input device would increase the input bandwidth, this has to be traded-off with an increase in input manipulation complexity.

6.8.2 Alternative Selection Techniques

Along with testing other input devices, future work could also explore other selection techniques. We explored variations of the ray casting metaphor, but did not explore enhancements to the hand extension metaphor. One possible enhancement would be to explore 3D volume cursors, such as the silk cursor [Zhai et al. 1994], which have increased activation areas. While this would reintroduce the problem of selection ambiguity, a 3D extension of the bubble cursor [Grossman and Balakrishnan 2005a], which is a 2D area cursor that disambiguates between multiple targets by changing its capture area dynamically, could be used. One possible drawback of the technique is that it could suffer from the same increased input device footprints which we observed for the point cursor. Indeed, some initial studies in VR indicate that such a technique would not perform as well as the depth ray [Vanacken et al. 2007]. However, its success in 2D environments does warrant the exploration.

Another new technique which could be explored is bimanual ray casting selection. As discussed above, such a technique would use both hands, to each cast a ray, and the selection point would be defined as the midpoint of the shortest line connecting to two rays. This technique could also be used to define a selection volume. The midpoint of the line connecting the two rays would instead be used to define a spherical volume center, and the diameter of the sphere would be the distance between the two rays. This would be like a 3D bubble cursor, except instead of the radius being updated automatically, users would have explicit control over the selection volume. This could be useful if the user wished to select multiple targets, which is not permitted with the bubble cursor. We further investigate multiple target selection in Chapter 8.

Along with multiple target selection, it might also be useful to determine how our techniques could be applied to moving selection or tracking tasks. In such scenarios, techniques which use direct hand mappings may perform better, especially those which increase the cursor activation area, such as the silk cursor [Zhai et al. 1994].

Finally, it would be useful to consider how our techniques could be used for object manipulation. Although the depth ray was found to be efficient for selection, translating an object with a ray casting metaphor may be awkward in comparison to using a directly mapped 3D cursor [Bowman et al. 1999]. A potential method would be to combine the depth ray, for the selection phase, with direct hand mappings, for the manipulation phase, similar to Bowman's HOMER technique [Bowman et al. 1999]. We explore this combination of selection and manipulation in Chapters 7 and 8.

6.9 Conclusions

We have presented an in-depth exploration of selection techniques for volumetric displays. In a first experiment, we found that the ray cursor is significantly faster than the point cursor for single target environments. This is consistent with evaluation of selection techniques on other 3D display platforms, despite the within-reach size of volumetric displays. Furthermore, this study confirmed our own work from Chapter 5, which indicated that a 3D point cursor would be difficult to use due to the added perceptual and dimensional constraints.

Based on this result, we were motivated to design enhancements to the ray cursor technique, to provide disambiguation mechanisms such that the new techniques would be suitable for dense target environments. We presented four design alternatives, each with their own unique properties. In a second experiment, we quantitatively evaluated the benefits and drawbacks of each of our new techniques.

The most successful technique was the depth ray, in which users selected and disambiguated their target somewhat concurrently. The technique significantly lowered acquisition time, input device footprint, and error rate, in comparison to the 3D point cursor. The lock ray and flower ray also performed well, both reducing input device footprint and error rates, but their acquisition times were not as good.

In summary, we have provided important data on 3D selection techniques for volumetric displays, including new techniques which we have designed which reduce selection times, error rates, and input device footprints. We have discussed the implications of our work to future interface design, and possible extensions to our work. These contributions will be valuable for future designers of interactive volumetric display applications, as object selection will be a fundamental technique for any such application.

With this core operation now thoroughly understood and investigated, we will now turn our attention to higher level user interface design issues. While the other "universal" 3D interface operations (system control, manipulation, and navigation), could be studied independently, as selection was in this and the previous chapter, we felt that to fully understand the issues surrounding these higher level interface operations, it would be more useful to explore them within working user interfaces. We will do just that in the next two chapters. In Chapter 7, we explore a user interface for volumetric displays where a single user interacts with the system with direct touch and multifinger gestures. In Chapter 8, we explore a collaborative user interface, and its associated interaction techniques.

7. User Interface Design

"Any baseball is beautiful. No other small package comes as close to the ideal design and utility. It is a perfect object for a man's hand. Pick it up and it instantly suggests its purpose; it is meant to be thrown a considerable distance - thrown hard and with precision."

- Roger Angell

7.1 Introduction

[1][2]In the previous chapters we have discussed the unique properties of volumetric displays, and investigated the implications of these properties on low-level human factor issues related to perceptual and motor-skill abilities. We have also designed and evaluated new techniques for object selection in volumetric displays, which will be a fundamental task in almost any user interface. Such explorations can be seen as laying the groundwork for our high-level goal of using volumetric displays as interactive platforms. However, other operations, such as object manipulation, system control, and navigation, remain to be explored. These tasks are generally higher-level in nature, having a tighter integration with the user interface which they are implemented into. Furthermore, the quality of such techniques will be much more subjective in nature, and so low-level quantitative studies, such as the ones we have carried out for object selection, will be less appropriate. As such, we will explore such techniques, within actual working user interfaces. This exploration will involve the integration of low-level tasks such as object selection, with higher level tasks, such as browsing menus, accessing

[1] A preliminary version of this work is published in Grossman et al. [2004].

[2] This Chapter is supplemented by Video Figure 7-1, available online at: http://www.dgp.toronto.edu/~tovi/thesis and https://tspace.library.utoronto.ca/handle/1807/9944.

widgets, and manipulating data. In essence, we wish to explore what the equivalent to the traditional desktop graphical user interfaces may look like on a volumetric display.

In this chapter, we will investigate these interaction techniques through the development of an interactive 3D geometric model building application. While this application area itself presents many interesting challenges, our focus is on the interaction techniques that can likely be generalized to interactive applications for other domains. We explore a direct style of interaction where the user interacts with the virtual data using direct finger manipulations on and around the 3D display enclosure (Figure 7-1).

Figure 7-1. User working with a volumetric display, with finger input tracked using a camera-based motion tracking system.

7.2 Related Work

Much research done on interactive 3D virtual environments to date has relied on stereoscopic displays, either immersive VR systems [Buxton and Fitzmaurice 1998], or non-immersive fish tank VR systems using LCD shutter stereo-glasses [Ware and Balakrishnan 1994, Ware and Lowther 1997, Zhai 1995]. Given that volumetric displays have not been easily available until recently, there has been relatively little research on how to use such displays effectively in an interactive manner. A speculative paper [Balakrishnan et al. 2001] discusses possible interaction scenarios for volumetric displays, using wizard-of-oz mock-up prototypes to demonstrate various techniques for selection, displaying text and menus, and manipulating objects. However, they did not have or make use of a real volumetric display and as such did not demonstrate any working implementations of their ideas.

From an interaction perspective, the most relevant prior art is in the virtual reality community, that has long explored interactive 3D environments, albeit not with true 3D display technologies. This includes work on virtual object selection, virtual object manipulation, menu and command selection, and various 3D widgets. In Section 2.6 and 2.7 we provided an overview of this research in 3D interaction. Here we summarize the main results which are most relevant to work presented in this chapter.

In many virtual environments, (e.g., [Liang and Green 1994]), objects are manipulated using a six degree-of-freedom tracker. This approach allows for straightforward mappings where the position and orientation of virtual objects correspond directly to the tracker's movements. Others [Mine 1995b] have used direct gestural interaction where hand movements are mapped directly to object movement. The HOMER technique [Bowman and Hodges 1997] combines ray-casting selection with subsequent direct manipulation: after an object is selected, its position and orientation is manipulated as though it were attached to the hand directly. In Charade [Baudel and Beaudouin-Lafon 1993], freehand gestures were used to manipulate 2-dimensional computerized objects in an augmented reality system.

Conner et al [1992] present a set of 3D widgets that allow for indirect interaction with virtual objects through a mediating virtual widget with clickable elements. For example, a translation widget would have virtual handles representing the three primary axes that could be dragged to move the corresponding virtual object in that direction. Many current applications (e.g., MAYA, 3D StudioMax) for 3D modeling and animation make extensive use of such 3D widgets since they can be easily operated with status-quo mouse & keyboards input.

In this chapter we will discuss the use of a two dimensional menu placed on the surface of the display. A similar idea was included in [Mine 1995b], where a 2D menu was embedded in the virtual environment. The menu they developed floats in 3D space and includes various widgets such as radio buttons, sliders, and dials. The user interacts with the menu using a ray cursor, so that the user does not have to make large reaching movements. In our implementation, we place the menu on the surface of the display so that the user can also directly reach and interact with it. In the JDCAD system [Liang and Green 1994] a ring menu was used for item selection, where the items were arranged along the circumference of a circle, and could be rotated until the item to be selected was directly in front of the user.

In short, there exists a significant body of work in the general area of 3D interaction that we can build upon in our designs for interactive volumetric displays. However, little of this prior art is directly related to volumetric displays per se. In Section 1.2, we discuss how the unique properties of volumetric displays introduce new interaction challenges which are not necessarily addressed by the previous literature. In particular, volumetric displays provide a fixed display area around which to center interactions, which make it fundamentally different from traditional virtual environments. Thus, interfaces specific to volumetric displays is a ripe area for further exploration.

7.3 Design Guidelines

We identified several interesting design guidelines for user interface design which felt would be important to explore. These guidelines were identified based on our experiences with volumetric displays, acquired through the studies described in the previous chapters of this thesis, informal usage observations of users viewing static imagery on the display, and the unique affordances of volumetric displays, as described in Section 1.2.

7.3.1 Use the Volumetric Display as Sole Input and Display Platform

A volumetric display could be used as an output-only device to display 3D imagery that is created and manipulated using traditional 2D computational environments. In this usage scenario, the volumetric display will indeed enable users to better view a 3D scene, but it will be a passive viewing experience, much like watching a movie. This thesis is motivated by our belief that the enhanced 3D viewing capabilities of volumetric displays make it imperative that we begin to explore using it not only to view 3D images, but to also create and interact with those images directly on the volumetric display itself.

Thus, in this chapter, we focus on a user interface where the volumetric display is the exclusive platform for doing all manipulations with the displayed 3D data. It is critical that we understand the issues surrounding interaction with this class of display in isolation, before attempting to possibly integrate it into environments with multiple heterogeneous displays each with their own strengths and weaknesses.

7.3.2 Support Multi-Viewpoint and Out-of-Viewpoint Operation

On 2D or stereoscopic displays, users have a single viewpoint of the 3D scene at a given time. As a result, users have to rotate the scene frequently to view the parts occluded from the current viewpoint, or to enhance depth perception through motion. Head tracking can enable more fluid viewpoint changes, but only within the limited range of the display's field of view. Furthermore, most interaction occurs relative to the current viewpoint. As we have discussed in Section 1.2.3, a volumetric display allows users to walk around it, or move their heads appropriately, to dynamically adjust their viewpoints in a fluid, unobtrusive manner much like they would when looking at a physical object in the real world. Furthermore, users can also reach around and interact with the 3D scene from all directions around the display, regardless of their current viewpoint (assuming a moderately sized display). These properties can allow for new interaction techniques beyond what is possible in other display environments and should be exploited to maximal benefit. We attempt to leverage these properties where appropriate in the design of our interaction techniques.

7.3.3 Utilize Direct Touch and Gestural Input

One can imagine using many possible input devices for interaction with volumetric displays. In Chapters 5 and 6 we used a six degree-of-freedom tracker to control a variety of selection cursors, and in Section 6.8.1 we discussed a number of alternative devices which could be used. The nature of the display, with 3D imagery floating within the enclosure, tends to evoke a strong tendency for people to touch it. Indeed, we have observed countless visitors to our lab attempting to point to parts of the displayed 3D scene by touching the surface of the display's enclosure, or gesturing with their fingers over it. This anecdotal evidence suggests that direct touch and gesture based input could be particularly suited as an input modality to enable rich, high quality, interaction. We explore this style of input throughout the interface design in this chapter.

7.4 System Hardware & Software

7.4.1 Display Device

We used the same 3D volumetric display that was used in the studies described in the previous chapters. A full description of the display is provided in Section 1.4. This generation of display technology possesses a number of viewing artifacts, as described in Section 3.4. The resulting level of quality prevents the implementation of an application that could be used in real-world scenarios. However the viewing quality does not hinder our ability to investigate important user interface issues.

7.4.2 Finger Tracking

A Vicon motion tracking system (www.vicon.com) is used to track the positions of markers placed on the user's fingers. The Vicon system uses several high-resolution cameras to track the 3D location of multiple passive reflective markers in real time. In addition to tracking the location of the markers in 3D space, the system can uniquely identify and label each marker according to its position on a user's fingers. The 3D coordinates of these labeled markers can then be streamed in real-time to other applications. Our prototype uses four cameras for tracking, two of which are seen in Figure 7-1. In our current work, we track markers on the index fingers of both hands, and the thumb of the user's dominant hand (Figure 7-2). This allows the user to perform some more complex gestures with the dominant hand.

We use the labeled marker data, in conjunction with knowledge of the precise topology and 3D spatial location of the display's enclosure in the tracking volume, to simulate an enhanced touch sensitive display. Our system categorizes the precise positional information of the tips of the two index fingers and thumb into one of three discrete states: "down" – when touching the surface of the display's enclosure, "hovering" – when within 6 cm of the surface, and “up” – when more than 6 cm from the surface. We also detect static postures and dynamic gestures of the fingers by examining the relative distance between the markers.

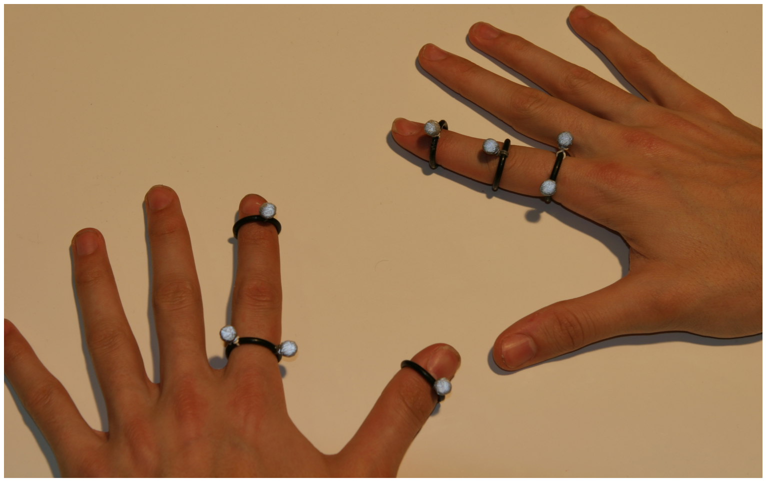

Figure 7-2. Markers used for tracking finger positions. Multiple markers enable us to track bending of the index fingers, as well as finger tip position and orientation. A marker is placed on the dominant hand thumb to support additional gestures with that hand.

Note that it is technically possible to make the display’s enclosure directly touch sensitive with current transparent resistive overlay technology, but the cost would likely be prohibitive given low production volumes. As such, we use this motion tracking system to simulate a touch sensitive display surface. Furthermore, this tracking system allows us to explore postural and gestural input that would not be possible with only a simple touch sensitive overlay. We also note that while current generation tracking technology requires markers for robust tracking, improvements to computer vision techniques may reduce or possibly even eliminate the need for markers in the future. While the inconvenience of using markers does marginally detract from the overall usability of our prototype system, this tracking system allows us to explore advanced freehand interaction techniques today, *before* marker-free tracking becomes widely available. As such, this hardware setup should be viewed simply as an enabling technology for our prototype, rather than one that would be used in any future real implementation of our interface ideas.

7.4.3 Software

Our application software was written in C++ and OpenGL, with a custom OpenGL driver specific to the volumetric display. Marker tracking and labeling was performed using Vicon's standard tracking software, and the data streamed in real-time to our application. The markers were tracked at 120Hz. We could not detect any perceptible latency in the marker data or in the movement of virtual finger representations relative to the actual finger movements.

7.5 Command Input

Given that we intend to perform all interaction on and around the display itself, and wanted to avoid using additional input devices like keyboards, we implemented two techniques to facilitate command input using the fingers: *surface menus* and a set of *postures and gestures.*

7.5.1 Surface Menus

Similar to interfaces for 2D touch screens, we display frequently used commands as buttons on the surface of the display. We call these *surface menus* (Figure 7-3). Because there is 3-10 cm gap between the edge of the display volume and its enclosing surface, these options could not be digitally displayed directly on the touchable enclosure. Since this gap will likely not be present in future implementations, we felt it was reasonable to simulate the buttons using pre-printed acetate overlays (Figure 7-3). We provide two surface menus, one for each hand. The buttons on the non-dominant hand's surface menu are used for kinesthetically held transient modal commands. These provide context for the dominant hand's interaction with the system while the non-dominant hand's index finger is down on the appropriate button (much like the use of a "shift" key in a regular keyboard). The buttons on the dominant hand's surface menu are for other frequently used commands, and are executed with a quick tap.

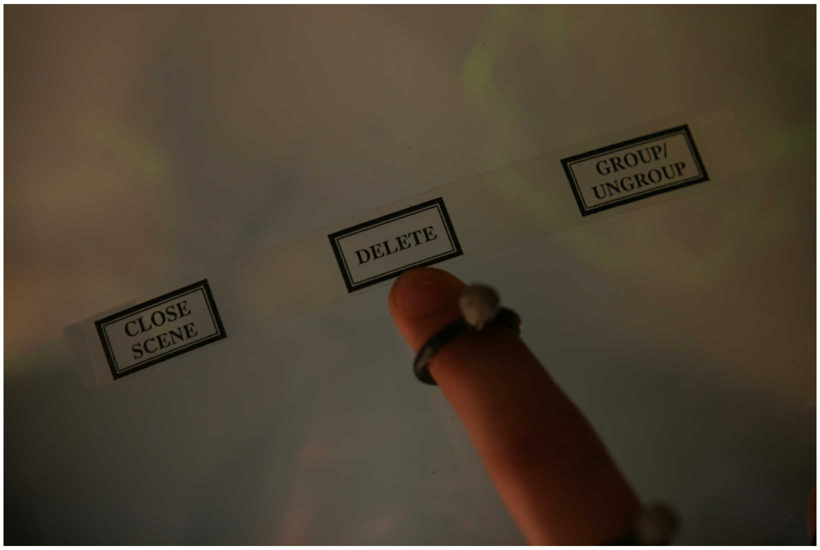

Figure 7-3. Surface menus. A physical transparency taped onto the display's surface provides buttons which can be tapped to execute a function or tapped-and-held to maintain a mode.

When either index finger hovers over any surface menu button, a caption associated with its function is displayed to the user, similar to the bubble help in conventional GUIs.

In our implementation, the locations of the surface menus were fixed. However, if such menus were to be displayed digitally, their location could be dynamically adjusted. It would be desirable to orient the menus relative to the user's body position, such that they would always appear on either side of the user for easy access. We implement dynamic orientation for the button captions, such that the caption is always facing the user, regardless of the user's position around the display. The user's position is estimated by examining the position and orientation of their fingers.

7.5.2 Postures and Gestures

While surface menus provide a nice mechanism for command input, there are instances when it could be inconvenient to have to touch the surface menu buttons to invoke a command. For example, if the user is manipulating a virtual object, it may be easier to enter commands using other finger movements. We also wanted to experiment with more than one command input mechanism, to enable later determination of optimal solutions. Accordingly, we developed a set of hand postures and gestures which can be carried out on or off the surface of the display. We infer the set of postures based on the shape of the fingers, while the set of gestures is determined based on the dynamic characteristics of the fingers' movement over time. Figure 7-4 illustrates this set of postures and gestures. The commands associated with each posture and gesture will be described as we progress through the chapter explaining the various interaction techniques.

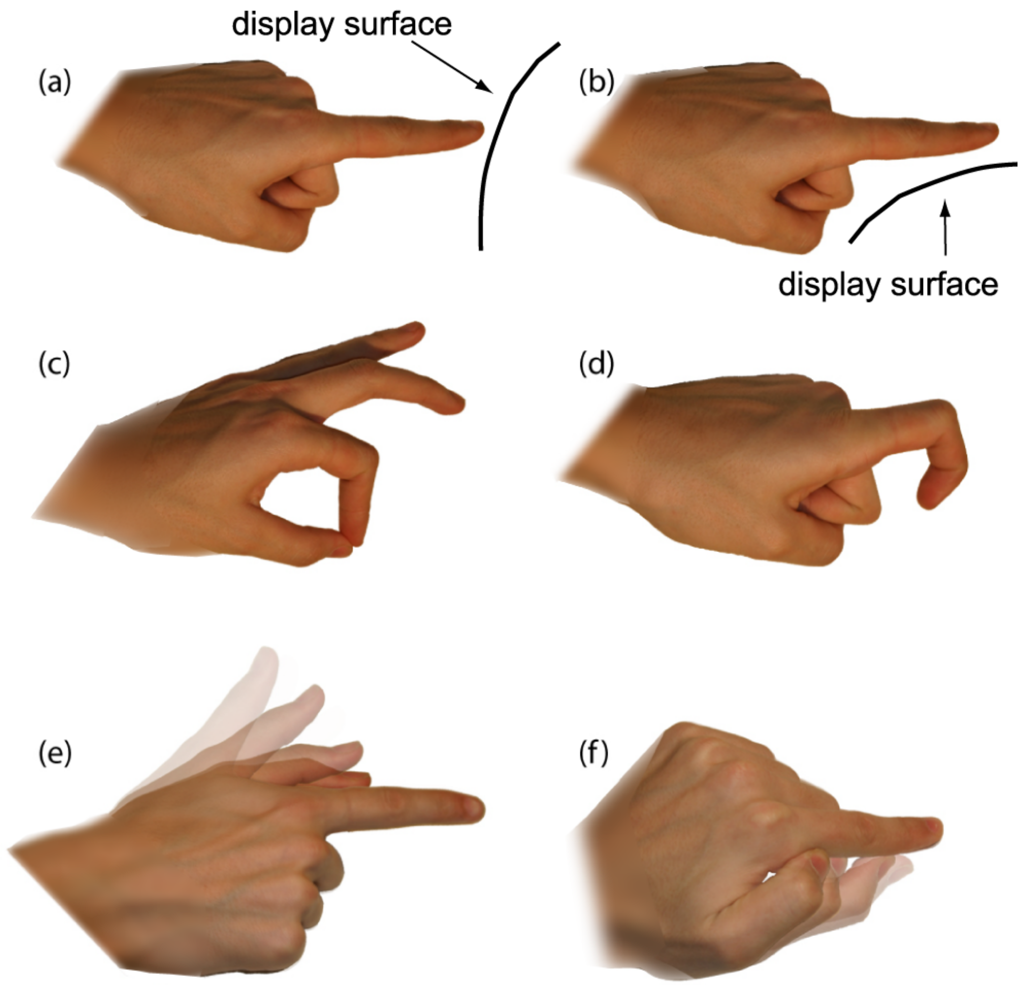

Figure 7-4. Postures and gestures. (a) point posture: index finger points towards the display. (b) flat posture: index finger is parallel to display surface. (c) pinch posture: tips of index finger and thumb brought together. (d) curl posture: tip bent towards base of finger. (e) trigger gesture: thumb presses against index finger (f) scrub gesture: thumb scrubs along index finger in either direction.

7.6 Interaction Techniques

For interacting in three dimensions, we need to support a variety of basic operations such as file visualization and browsing, selection, translation, scaling, and rotation. The following techniques were developed in the context of a 3D model building application, but can be generalized to other volumetric display applications. We tried to explore techniques that would be most important for future user interface design with volumetric displays, such that our work would have maximum generalizability. While some of these techniques are inspired by previous 3D user interface research, most have been adapted to be suitable for volumetric displays, based on their specific affordances.

7.6.1 SurfaceBrowser

In order to allow for basic file operations of load, save, organize, copy, and delete we developed a simple file/object management mechanism called *SurfaceBrowser* (Figure 7-5). The SurfaceBrowser displays various objects by organizing them into cells of a 2D array. Four such arrays, or *pages*, are then projected around the entire inner surface of the display, allowing the user to easily interact with the objects by touching the surface of the enclosure directly above them. The pages either contain *models* or *scenes*. Models are primary shapes used in building more complex scenes. The contents of each cell rotate slowly to aid in their visualization. We have currently implemented display support only for 3D models and scenes since they are the primary data types of interest in our 3D model building application. However, support for other data types such as images could easily be implemented within the same SurfaceBrowser framework.

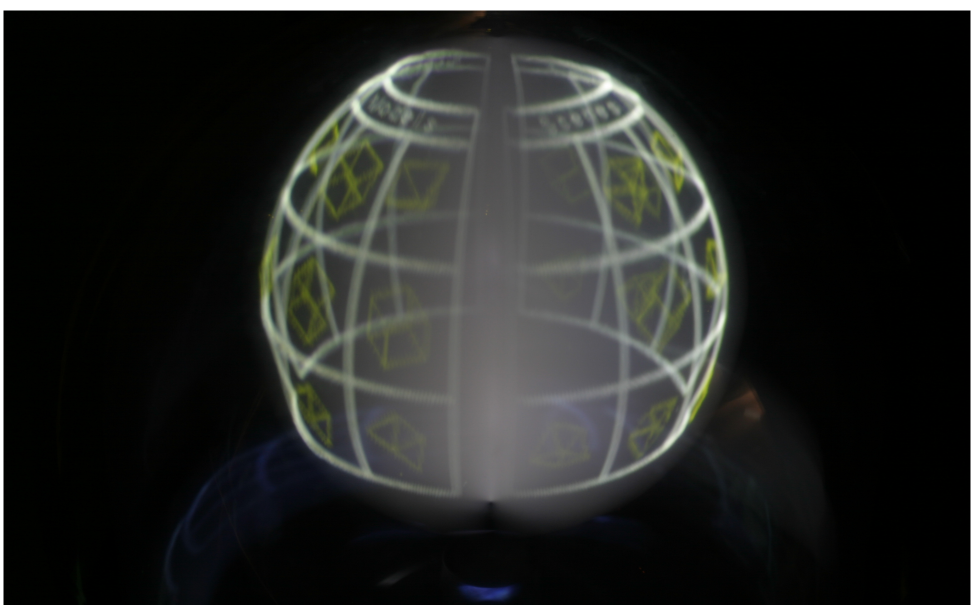

Figure 7-5. SurfaceBrowser. Four pages, each a 3x3 array of cells are projected around the outer surface of the display. The distortion at the centre of the display is an artifact of the physical display mechanism.

While the SurfaceBrowser is displayed, it can be rotated by scrubbing the non-dominant hand's index finger along the surface of the display. This feature allows the user to bring regions of interest closer to him/her, although this is not strictly necessary in a volumetric display since the user could walk or move their head around the display to look at various parts of the SurfaceBrowser. In a sense, this rotation technique supports "lazy" operation, which may be desirable in some situations.

A flat arrow cursor is displayed below the dominant hand's index finger, which is used to perform basic operations with objects in the SurfaceBrowser. Touching the surface with this finger while the cursor is above an object (either model or scene) selects it, lifting the finger deselects it. While selected, an object can be moved from cell to cell by dragging within a page, or copied by dragging to another page. The object can also be dragged into a trash can area at the top of the display to delete it. A quick tap on any object opens it. When this occurs, the selected model or scene smoothly animates from its 2D form on the surface of the display, to its 3D shape at the center of the display. We use smooth animated transitions throughout our prototype to provide users with a sense of continuity as they move from action to action.

7.6.2 Model Transformations

Once a model has been opened, we allow for rotation, translation, and scaling. While we did explore simultaneous rotation and translation for six degree-of-freedom manipulations, we limited the interactions to distinct modal operations to enhance precision. We now describe the interaction techniques that allow 3D transformations to be applied to the models.

7.6.2.1 Rotate

Rotation is initiated by touching the display with the dominant hand's index finger. The finger is then dragged across the surface of the display, and this movement is transformed into rotation of the model, as if there were a stick connecting the finger to the model's center (Figure 7-6a, b). This provides two degrees of freedom for rotation. A third rotational degree-of-freedom is achieved by twisting the hand, while the index finger is still down. This rotates the model about the vector defined from the finger to the model (Figure 7-6c). Rotation stops when the finger is removed from the surface of the display.

Note that we deliberately chose rotation axes that are defined by the display's hemispheric surface and the user's hands, in order to keep the mappings simple. Although our hemispheric volumetric display does have a well defined up-down axis, it does not have any inherent left-right or front-back axes. As such, although we do define a global three axis coordinate system (with admittedly arbitrary choices of left-right and front-back axes) that we use for snapping operations described later, we did not want to impose this global axes triad on the user for the basic transformation, particularly since the user could be performing these transformations while standing anywhere around the display.

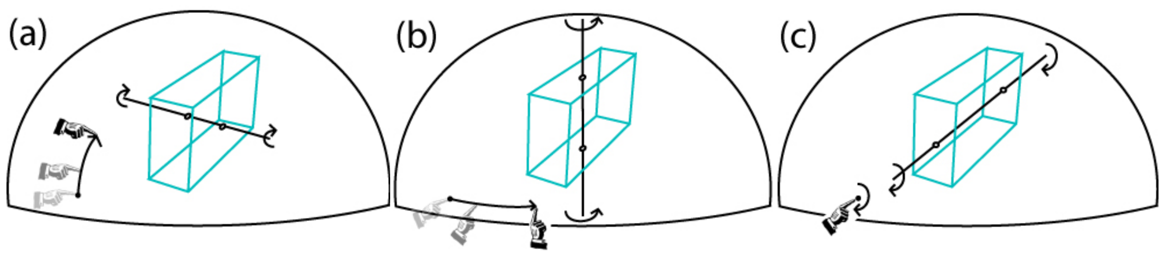

Figure 7-6. Rotating a model. (a, b) Moving finger across the surface rotates the model as though its centre was attached by a stick to the fingertip. (c) Twisting the hand about the fingertip rotates the model about the vector from its centre to the fingertip.

7.6.2.2 Translate

Translation is imitated by assuming a pinch posture (Figure 7-4c) with the dominant hand. While pinched, moving the hand in any direction moves the model the same distance in that direction. The metaphor here is that of picking up an object with a pinch grip and moving it. When translating the model away from the user, it is possible that the user's hand may collide with the display. Thus, we provided added functionality to translate the model towards or away from the user in a relative manner, accomplished through the scrub gesture (Figure 7-4f). The direction of the scrub gesture determines the direction of the model's translation along the vector defined by the index finger. When the dominant hand leaves the pinch posture, and is not scrubbing, translation stops. Since entering or exiting the pinch posture can cause unwanted translations, we provide a clutching mechanism to freeze the model momentarily to ensure precision translations.

This freezing action occurs whenever the non-dominant hand's index finger is in the curl posture (Figure 7-4d). The metaphor is that of curling the fingers around an object to hold it still, as one might do in the physical world.

7.6.2.3 Scale

Unlike translation and rotation which are performed with a single finger, scaling is a bimanual technique. To scale a model, both index fingers are placed on and dragged along the surface of the display. Sliding the fingers further apart on the surface increases the scale, while sliding them together decreases it. The object is scaled uniformly along all dimensions. This is similar to the technique presented in [Kurtenbach et al. 1997] for scaling in a 2D drawing program. Scaling stops when the user lifts either finger from the surface.

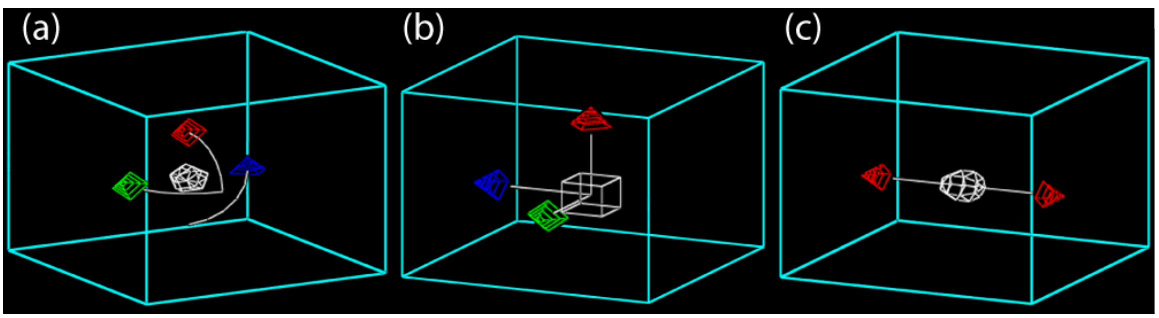

Figure 7-7. Visual feedback for transformations. (a) Rotate. (b) Translate. (c) Scale

7.6.3 Visual Feedback for Transformations

While performing any of the transformations, a colored 3D icon is drawn at the center of the model, indicating which transformation is currently being applied (Figure 7-7). The icons are displayed oriented towards the user's current hand positions to facilitate viewing. If the transformation has not yet been initiated, the icon is white, indicating to the user that the posture of their hands is close to that required to begin the corresponding transformation. For example, if both of the user's hands are close to touching the surface of the display, the scale icon will be displayed in white. Once both fingers make contact with the surface the widget will become colored. This provides a nice way to guide users into appropriate postures for transformation actions, and also to reduce accidental triggering of transformations.

7.6.4 Constrained Transformations

The rotate and translate transformations discussed allow users to simultaneously control three degrees of freedom. In most common 3D graphics applications, such transformations are limited to one or at most two degrees of freedom primarily because they are performed using 2D input devices within a two-dimensional perspective viewpoint. While allowing users to simultaneously move or rotate objects along three axes simultaneously on a volumetric display is very powerful, it is sometimes useful to constrain transformations to a particular axis for precision movements. To support this, we created a mechanism where users can add or remove axes to which subsequent transformations will be constrained.

7.6.4.1 Axis Definition

The constraint axis specification mode is entered and maintained while the non-dominant hand's index finger is held on the "axis" surface menu button. While in axis mode the dominate hand can create, activate, or deactivate constraint axes. When in a pointing posture, a white constraint axis preview line is displayed as the finger hovers over the surface of the display. The position and orientation of the preview line matches the vector of the finger, so it appears as though it is a ray being emitted from the finger tip. If the preview line is close enough to one of the global primary axes, or the center or object axes of any models in the scene, it will snap to that axis. This aids in precise positioning of the constraint axis. If the finger taps the surface while still in the pointing posture, the constraint axis will be added, and is displayed as a thick red line extending through the display (Figure 7-8). Tapping either end of the constraint axis will deactivate that constraint axis, and it will appear as a short white tick mark. Multiple deactivated constraint axes can exist at a time. The tick marks will appear as long as the non-dominant hand maintains the system in axis mode, and tapping any of them in the flat posture will activate that constraint axis. The active constraint axis is deactivated if another constraint axis is created or activated, so that only one constraint axis can be active at a time.

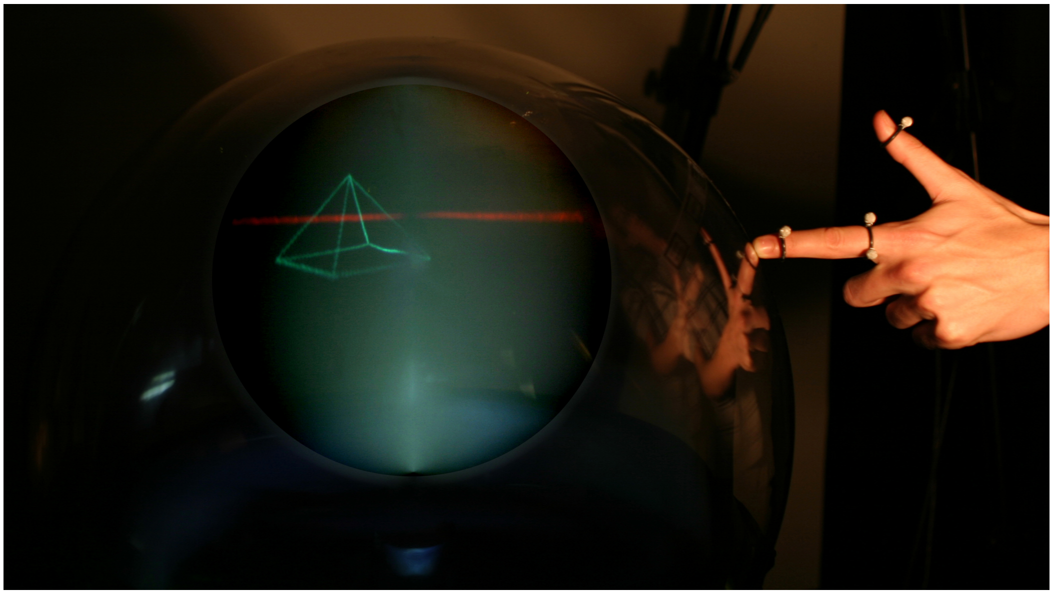

Figure 7-8. Constraint axis. These can be placed arbitrarily by pointing to define the axis and touching the surface of the display to add it to the scene.

7.6.4.2 Constrained Transformations

If a constraint axis is active, then all transformations will be constrained to it. For rotation, only the component of the movement of the index finger that is perpendicular to the constraint axis is applied to the rotation of the active model. This causes the model to rotate about the defined axis. A large cylindrical widget is drawn perpendicular to the constraint axis, providing feedback for the user as to where their finger should be dragged for effective rotation. Similarly, when translating, only the component of movement parallel to the constraint axis is used. The translation is thus constrained to that axis. By default our scale function is a one degree-of-freedom operation, where objects are scaled uniformly along all dimensions. However, when a constraint axis is active, the scale operation is constrained to apply only along that axis.

7.6.5 From Models to Scenes

Now that the main interaction techniques for manipulating individual models have been described, we will discuss the techniques involved in combining multiple models to build up scenes.

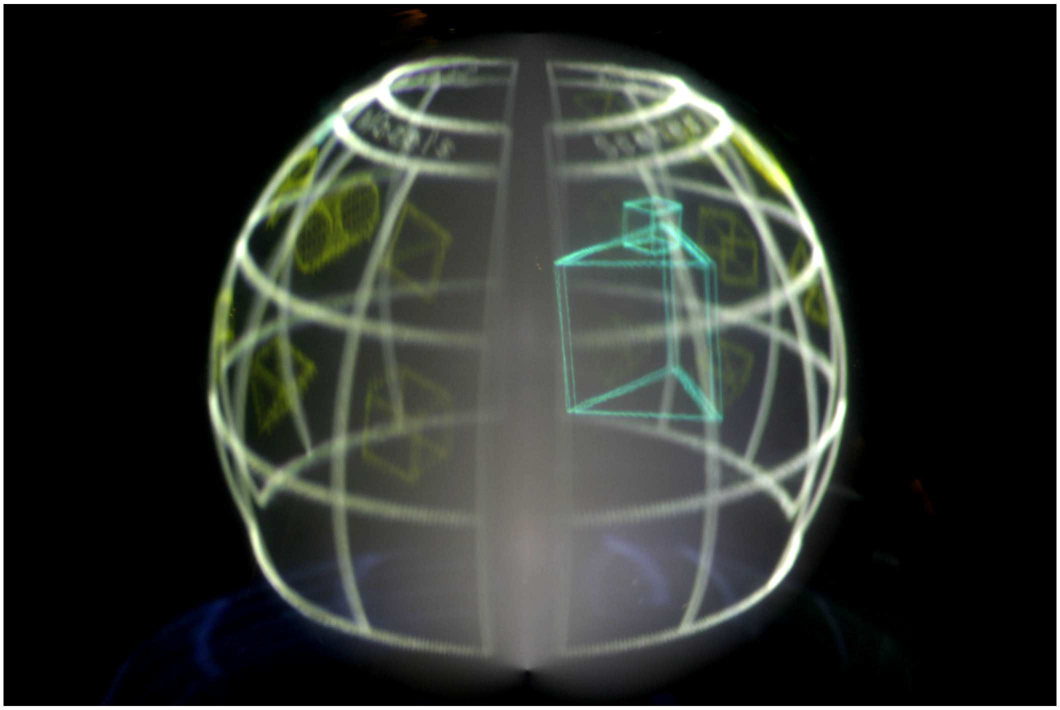

Figure 7-9. Adding additional models to the scene. The surface browser allows context to be maintained while making a selection that affects the scene.

7.6.5.1 Add model

Additional models can be added to the current scene by depressing the “Add Model” surface menu button with the non-dominant hand’s index finger. When this is done, the SurfaceBrowser is shown around the perimeter of the display, while the current scene continues to be rendered in the center (Figure 7-9). As always, while the non-dominant hand’s finger remains down, it can be dragged across the surface to rotate the SurfaceBrowser, and a model is added to the scene by tapping it. The SurfaceBrowser disappears when a model is added, or when the non-dominant hand’s finger leaves the surface of the display.

7.6.5.2 Selection

With multiple models in a scene, we must select which models are to be manipulated before performing any further transformations. We support model selections with a scene using the depth ray, which was found to be an effective technique for static selection within a volumetric display in Chapter 6. The depth ray can be used by tapping-and-holding on a surface menu button with the non-dominant hand. However, in this implementation, the ray appears to emit from the user's finger (Figure 7-10). The ray is rendered as a yellow line. When the ray intersects a model, the model is highlighted by changing color, and a trigger gesture (Figure 7-4e) is used to select or deselect it. Models turn blue once they are selected, as a visual indicator of their state. By using the depth ray, the user can indicate an object of interest when multiple targets are intersected. Moving the hand forward will highlight the model further from the hand, and moving the hand backwards will highlight the model closer to the hand. Subsequent actions affect only the selected models. It is important to note that our technical implementation allowed the user to control the depth ray while the hand is above the display surface. If the user's hand had to be touching the surface, it would become difficult to control the position of the depth marker. One possibility would be to control the depth through pressure against the surface.

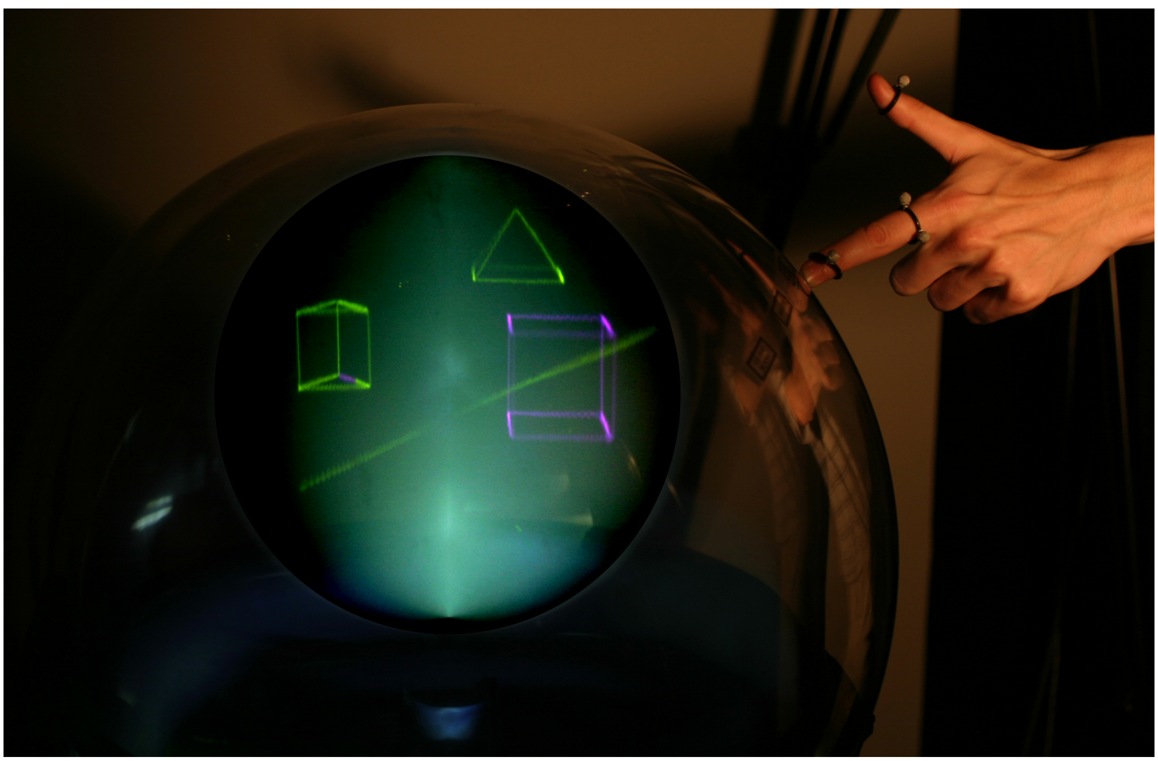

Figure 7-10. Selection using a ray cursor.

7.6.5.3 Operations on Multiple Models

The transformation techniques for single models described earlier can be used to simultaneously apply transformations to multiple selected models. When more than one model is selected, the center of rotation, translation, and scaling operations is computed as the average of the centers of all selected models. The relative positions of the selected models remain unchanged by any transformation.

Multiple objects can be grouped together by pressing the “Group/Ungroup” surface menu button when they are selected. Once a group has been formed, selecting or deselecting a model selects or deselects its entire group. Objects can subsequently be ungrouped by pressing the “Group/Ungroup” button while a group is selected.

Single models or groups of models, once selected, can be deleted from a scene by pressing the “Delete” button.

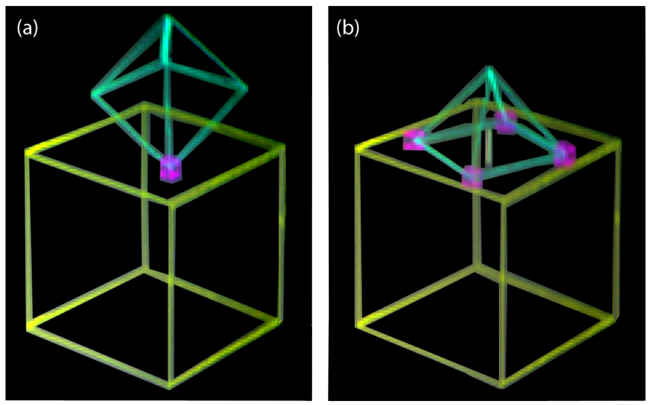

Figure 7-11. Snapping. (a) Purple marks indicate snapping of a vertex to a face. (b) White tick marks indicate snapping of parallel faces.

7.6.5.4 Snapping/Collision

Our goal was not to address the specific functionality needs of a geometric building application, but to develop general interaction techniques for volumetric displays. However, to gain a better understanding of the user interface, we did wish our interface to support basic model building operations, so that the interface could actually be used for a purpose. To aid in building complex scenes, the system supports collision detection and snapping between models. By default this is disabled, but by pressing the “Snap” surface menu button, collision detection and snapping are enabled. When translating, objects cannot intersect one another. Instead, the vertices of the moving object will snap to the face of any stationary object which it intersects. Purple marks are displayed to indicate the point at which a vertex snaps to a face (Figure 7-11a). If the model is then translated to any side of the face, it will snap to the edge. Objects can also snap while rotating. The object will snap such that the normal of any of its faces matches the normal of any nearby face. A small tick mark is displayed through parallel snapped faces to indicate that snapping has occurred. (Figure 7-11b).

7.7 Discussion

The interaction techniques we developed make maximal use of the 3D nature of the display and input system. While the direct finger tracking allows users to perform high degree-of-freedom operations with multiple fingers, the display technology allows users to accurately visualize the virtual 3D manipulations with excellent depth perception. Taken together, this allows for interesting interactions not possible in traditional input and display combinations. For example, free-form transformations allow for simultaneous three degree-of-freedom translation and rotation, allowing quick and accurate object placement in a 3D environment without the need for constant viewpoint rotations. Combining this with snapping and collision detection, users can quickly build-up 3D scenes from a set of primary models (e.g., Figure 7-12). To improve precision, users can quickly define arbitrary constraint axes. Not only can the constraint axes be defined from any viewpoint, but they can also be adequately visualized, without the need for adjusting a camera position as would be required in traditional 3D virtual environments.

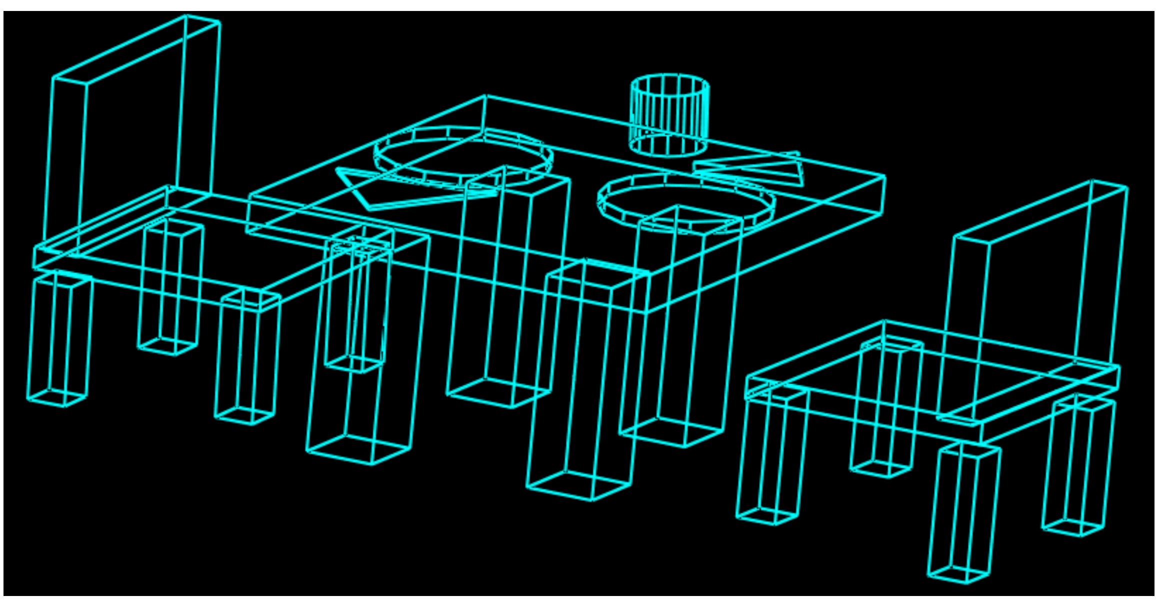

Figure 7-12. A "table setting" scene built using our system. The scene is rendered in high quality on a standard display to show the precise composition of its parts.

While some of the interaction techniques which we implemented have been adapted from research on virtual reality systems, others have been designed specifically for volumetric displays. For example, the ray casting metaphor has been demonstrated previously in VR literature, and in our own work in Chapter 6, but we adapted the technique to use the index finger to directly control the ray's orientation and position as well as using the thumb to perform a trigger gesture to confirm selections. Similarly, we extended previous work on virtual rotations to a new technique that enables precise rotations about all three axes simultaneously by moving the fingers directly on the surface of the display.

While the model manipulation techniques make use of up to three degrees of freedom of input, our mechanism for selecting between models and scenes, the SurfaceBrowser, is constrained to the inner surface of the volumetric display. This, in effect, creates a 2D viewing plane, wrapped around the inside surface. We deliberately designed the SurfaceBrowser in this manner, rather than making use of the full 3D display volume, because it frees up the internal display area for simultaneously viewing models selected from the SurfaceBrowser. Thus, a user is able to view their current work area and make selections from a menu that affects that area, preserving context. Because our volumetric display has a non-uniform gap between the surface and the rendered 3D volumetric image, we felt it was necessary to add a cursor icon to give the user feedback as to where their finger position was being mapped into the surface browser. As volumetric display technology improves, this gap will quite likely be reduced, and the cursor could be removed from the SurfaceBrowser, resulting in a sense of even more direct interaction.

It is important to note that although the techniques we developed were presented in the context of a 3D model builder, they are equally applicable to any interactive 3D application which uses a volumetric display. In a sense, our techniques are 3D analogues of the standard WIMP techniques used on 2D desktops that work reasonably well for a broad range of 2D applications. For example, the SurfaceBrowser can be thought of as a standard file browser for managing any kind of 3D, or even 2D, data on a volumetric display.

7.8 Implications for User Interface Design

Our work has direct implications for the design of user interfaces for volumetric displays, as the majority of the interactions which we have developed can be generalized to almost any application domain. Furthermore, in our research, we uncovered several interesting principles and issues unique to the development of user interfaces for volumetric displays. These properties have important implications for the future development of user interfaces.

7.8.1 Potential to Couple Input and Display Space

Our observations indicate that users wish to interact with imagery displayed on volumetric displays by directly touching the surface. Implementing direct touch input removes a layer of abstraction between input and display space, and thus tend to better afford gestural interactions. Similar observations and recommendations have been made for other display forms, such as tabletop displays, which also afford direct touch [Wu and Balakrishnan 2003].

In our work we developed a number of interactions through such direct touch gestures, which could inspire future interactions for volumetric displays. However, this implication must be taken with caution. Direct touch system may not always be appropriate. For one, they introduce a learnability problem, if the interface requires complex gestures which are not immediately obvious to the user [Baudel and Beaudouin-Lafon 1993]. Also, extended use of direct touch gestures may be tiring for a user. As such, the interaction metaphor which we explored in this chapter may be more relevant for casual or short-period usage scenarios. For example, a designer may approach a display to quickly see the latest versions of a model being developed in the studio.

7.8.2 Within-Reach Interaction

Unlike other virtual environments, the display space of volumetric displays is constrained by the size of the physical volume. This means that until large room-sized volumetric displays are developed, all objects will be within arm's reach. As a result, traditional 3D interaction techniques don't necessarily apply, necessitating the development of new techniques. For example, less attention is required for navigational issues, while more attention should be focused on overcoming the barrier which prevents users from reaching in and grabbing objects.

7.8.3 Hidden Surface Removal

As we have discussed, the current generation of volumetric displays do not support hidden-surface removal. While this did not hinder our efforts to explore the development of a user interface, there are likely usage scenarios where the user would desire hidden surface removal. It can be noted that if the location of the user is known, than the software could manually hide aspects of the imagery that should be occluded from the user's view. This only applies when there is a single user; extra considerations would be required when multiple users are viewing the imagery. In the next chapter, we will investigate this issue further.

7.8.4 Potential for Collaborative Usage

One of the important defining properties of volumetric displays is their 360° viewing angle. In this chapter we leveraged this property by designing multi-viewpoint and out-of-viewpoint operations. How this property also has important implications for the development of collaborative applications. In such an application, multiple users could stand around the display, and have their own unique perspectives of the displayed imagery. We thoroughly explore collaborative usage of volumetric displays in the next chapter.

7.9 Future Work

While the techniques which we have presented here can form the basis for highly interactive use of volumetric displays, there are clearly many more interesting research challenges that remain to be explored.

7.9.1 Advanced Manipulations and Gestures

With improvements to the display technology, it would likely be possible to display surfaced and textured models. When such rendering is available it would be of interest to explore more advanced object manipulations. For example, our existing techniques for directly interacting with the display surface could be built upon to perform various sculpting operations. Manipulations used when working with physical clay such as pulling, pushing, squeezing, and stretching could be adapted to gestural interaction on the display surface to perform various surface deformations on virtual models.

To allow for such advanced gestures, it may be useful to move from tracking the thumb and two index fingers which we implemented in our system, to full tracking of all fingers and the palm of both hands. This would clearly increase the possible set of postures and gestures. However there would be an obvious trade off. The small and simple gesture set which we developed allowed for a simple fluid interface requiring relatively little prior training of the few users who have tried the system. Increasing the complexity of the gesture set would require explicit mechanisms to reveal and teach novice users about the various possible interface actions.

7.9.2 User Testing

The work presented here contributes an exploration of user interfaces for volumetric displays. The system which we developed was not meant to be a working application, or to be used by any sort of end user. The system was more of a platform for our own explorations. However, that is not to say that there would be no value is obtaining feedback by performing user testing wtih the system. An important line of future work would thus be to have users try the system and provide feedback. This would provide us with initial feedback of our designs, and may also lead us to new design alternatives.

7.9.3 Multiple Display Environments

In this exploration we intentionally used the volumetric displays as the sole output device. This allowed all interactions to occur with the volumetric displays, and more importantly, allowed us to focus on the interaction issues specifically related to volumetric displays. However, in the future, it would be interesting to integrate the volumetric display into an environment with other display forms. For example, it may be interesting to provide the user with a flat interactive display surface in front of the volumetric display. This display could be used to provide 2D information and support 2D interactions, as in many cases, this may be more appropriate then interacting in the 3D volume of the display. For example, this could be an alternative to our surface browser.

Users could browse and select files using typical GUI interactions on the 2D display, and then when they wanted to view and interact with the model, it would be displayed in the volumetric display.

7.9.4 Alternative Input

We have explored user interface design for volumetric displays, where the means for input is through gestures on and above the display volume. This eliminates the need for auxiliary input devices, and also appeared to be appropriate based on the observed affordances of volumetric displays. However, as noted in Section 7.8.1, there may be instances where this method of input is not appropriate. Along with possible learnability and fatigue problems, it is also difficult to execute discrete operations, such as button clicks, with free hand gestures. As such, alternative methods of input should be explored. In Section 6.8.1, we discuss several input variations which could be considered for selection, which could all also be considered for interface control. In particular, we explore the use of a 2-button six degree-of-freedom input device in the next chapter.

7.9.5 Collaborative Usage

As discussed above in our implications, the display's 360° field of view makes it an obvious platform for exploring collaborative multi-user interaction in 3D environments. Given that users would not need to wear head mounted displays or special glasses, they would be able to view the 3D data while maintaining the context of their surrounding environment and other users, facilitating human-human communication in conjunction with human-computer interaction. However, because the work area is a shared 3D display, unique issues arise when multiple users attempt to share the space. For example, at what orientation should text be displayed? What kinds of strategies can be applied to allow two or more people to work together on such a display? The rich literature on collaborative computing will provide guidance, but these prior ideas will have to be adapted and refined for appropriate use with this new display technology, a challenge we believe is worthy of further explorations, which we initiate in Chapter 8.

7.10 Conclusions

We have presented an exploration of user interfaces for 3D volumetric display. To allow for high fidelity direct user input, we demonstrated the use of a real-time motion capture system to simulate a touch-sensitive display surface, detect hover just over the surface of the display, and track user hand positions when away from the display. We believe that this is an interesting use of motion tracking systems, departing from its traditional use in offline motion capture to provide movement data for animation, games, and human movement analysis. We developed a number of new interaction techniques, and integrated them into a working user interface for a geometric model building application. This was a generic application, meant as a platform for exploring interaction techniques. We discuss more specific application scenarios, for which these techniques could be adapted to, in Section 9.6.1.

This work builds upon the lower-level studies presented in the previous chapters, and in particular, we were able to integrate the previously studied depth ray into the user interface. We discussed the implications of our work to future interface design, and also outline several important future lines of work. In particular, the exploration of interaction techniques and user interfaces for a collaborative usage setting remains to be explored. In the next chapter, we will initiate this exploration.

8. Collaborative Issues and Interaction

"The key to being a good manager is keeping the people who hate me away from those who are still undecided."

\- Casey Stengel

8.1 Introduction

[1][2]In Section 1.2, we discussed a number of properties of volumetric displays which make them particularly suitable for collaborative interaction. An important advantage is that users do not have to wear any supplementary hardware which might hamper collaboration efforts. Further, the displays typically have a 360° field of view, allowing multiple users to work with the imagery from anywhere around the display. In Chapter 7, we investigated user interface design for volumetric displays; however the scope of our discussion was limited to single user applications.

We envision scenarios where a volumetric display could be utilized in a collaborative setting, which motivate such an exploration. As an example, a group of students could examine a virtual model of an anatomy specimen which they were studying in a laboratory. The students could work together to identify critical areas of the structure, label areas which may be anomalous, and perform various browsing operations to reveal hidden features. Carrying out such tasks on a volumetric display, while leveraging its unique features, could potentially improve the users' understanding of the 3D data, as demonstrated in Chapter 3 of this thesis, and in other studies [Rosen et al. 2004].

---

[1] A preliminary version of this work is published in Grossman and Balakrishnan [2008].

[2] This Chapter is supplemented by Video Figure 8-1, available online at: http://www.dgp.toronto.edu/~tovi/thesis and https://tspace.library.utoronto.ca/handle/1807/9944.

Before developing applications for such scenarios, it would be useful to first obtain a base understanding of the associated interaction design considerations. Our work in the previous chapter will provide guidance; however, the design of collaborative applications will likely raise new issues, which merits its specific investigation. In this chapter, we initiate this exploration by developing a prototype collaborative 3D model viewing application that served as a platform for our interaction designs (Figure 8-1). The prototype allows multiple users to inspect, markup, and manipulate 3D scenes. As in our work in Chapter 7, the interaction techniques which we implemented were designed such that they would be applicable across various application tasks and usage domains. However, where the techniques in Chapter 7 were focused on general user interface control and object manipulations, the techniques in this chapter are more focused on the exploration and markup of 3D models. At the conclusion of this chapter we provide the results from initial usage observation sessions and interviews with application domain experts, which, in general, were encouraging.

Figure 8-1. The prototype for our explorations allowed users to collaboratively interact with the volumetric display. Users' viewpoints and input devices were tracked in 3D.

8.2 Related Work

In Section 2.8 we provided a general overview of collaborative user interfaces. In this section we provide a more detailed discussion on the research which is most relevant to the work presented in this chapter.

8.2.1 Single Display Groupware

A branch of computer supported collaborative work which has received recent attention is single display groupware (SDG) - infrastructure which supports collocated groups interacting with a shared display [Stewart et al. 1999]. A defining element of SDG is that users can interact simultaneously with the display, using their own input devices.

Early research in SDG began with the MMM system [Bier and Freeman 1991], which allowed multiple users to simultaneously interact with several common interface elements, such as menus and text editors. Stewart et al. [1999] discuss three properties of SDG which have particular importance to our work:

8.2.1.1 Shared User Interface

Interface elements must be accessible and able to handle simultaneous input from all users. This can be especially problematic in direct touch systems, as there may not be a central location that users can reach to access interface elements. For volumetric displays, this means that placing direct touch widgets on the display surface, as was done in Chapter 7 may not be appropriate when multiple users are present.

One solution is to use popup menus and widgets, which can always be accessed regardless of user locations [Shen et al. 2004, Wu and Balakrishnan 2003]. An alternative solution is to use a non-direct input device. However, it can be challenging to define a control display mapping for non-direct input when users are interacting from various viewpoints [Wigdor et al. 2006]. Absolute mappings, which we have used in the previous chapters to control a 3D point cursor, will not be appropriate, as the user position may diverge from the absolute control space. As such, our work will examine other possibilities.

8.2.1.2 Shared Feedback

SDG Systems must have the ability to communicate information to multiple users simultaneously, and also to individual users. If the users are working from various positions around the display, then simultaneously presenting information can be problematic as it can suffer from orientation effects [Wigdor and Balakrishnan 2005]. As we investigated in Chapter 4, with volumetric displays, this will be particularly problematic, as what appears forwards to one user could appear backwards to another user. Presenting information to individual users is also problematic, as it can cause interference to other users for whom the information is not directed [Zanella and Greenberg 2001].

8.2.1.3 Coupled Navigation

When a single user navigates to a different area of the application data, other users will either also be forced to navigate simultaneously, which may be unexpected or unwanted, or have their views obscured by the one user who is navigating. This problem generalizes to any interaction which can result in conflicts when carried out simultaneously, or when unwanted by certain users. Greenberg et al. categorize such issues as concurrency control problems [Greenberg and Marwood 1994]. Possible approaches to this problem are to use locking mechanisms [Greenberg and Marwood 1994], coordination policies [Morris et al. 2004], or to rely on social protocols [Greenberg and Marwood 1994] to prevent conflicting actions. We anticipate that these approaches can be used for volumetric displays.

While most SDG research has been conducted with more widely available 2D technologies, some has been conducted in the 3D realm. Maybe most relevant to our work is the Two-User Responsive Workbench [Agrawala et al. 1997], which allows two users to stand around a physical table and interact with a 3D image. The users wear stereo shutter glasses, which interleave different images for each user, allowing each user to have an individualized 3D view of the scene. This work focused on implementation details and specialized views. Our work will explore new interaction techniques.

In summary, there exists a solid groundwork of research in single display groupware, but fewer results which are relevant to the design interactive applications for volumetric displays. Furthermore, due to the unique properties of volumetric displays, developing a collaborative application is not as trivial as generalizing existing SDG research. New interaction techniques specific to volumetric displays will need to be investigated.

8.3 Design Goals

Our review of the SDG literature, in addition to our own observational evidence, indicate that new interaction techniques are required to address the issues associated with collaborative use of volumetric displays. The following are three design goals which we have identified as having particular importance in developing such techniques.

8.3.1 Location Sensitive Interaction

Since users can stand anywhere around a volumetric display, the user interface should be accessible from any location, or be “omnidirectional”. A similar design goal has been followed for tabletop displays [Shen et al. 2004], which also have a 360° viewing angle. However, users of volumetric displays may be standing and walking around the display, so discrete seating locations cannot be assumed, as they commonly are in tabletop applications [Ryall et al. 2004, Shen et al. 2004, Wu and Balakrishnan 2003]. Furthermore, since interaction cannot be truly direct, as users typically cannot reach into the display, a user at the front of the display may want to interact with imagery at the back of the display. Systems, therefore, should not make territorial assumptions, such as correlating display areas with viewing positions, as they can in tabletop applications [Ryall et al. 2004]. As a result, orientating data and widgets to the closest possible viewing location, which is done in tabletop applications [Shen et al. 2004], may not be appropriate. Due to these additional challenges, we will explore “omnidirectional” interactions techniques, which can be used from anywhere around the display, but also leverage knowledge of the user’s viewing locations.

8.3.2 Parallel Access

Providing parallel access is a recognized design goal in SDG applications [Stewart et al. 1999]. Since we wish multiple users to be able to work with the display simultaneously, the user interface should be accessible to all users at all times, and interaction techniques should be able to be carried out in parallel. This design goal provides an interesting challenge when it comes to navigation. As discussed in our related work section, *coupled navigation* is a central difficulty in SDG applications [Stewart et al. 1999]. In volumetric displays, the problem is increased since navigation is one of the core interactions in 3D applications [Bowman and Wingrave 2001].

8.3.3 Inter-user Understanding

The last design goal is that users should have an awareness and understanding of what other users are doing. This property has been identified previously as “awareness” and is generally addressed with simple techniques such as cursor coloring [Bier and Freeman 1991]. However, the unique properties of volumetric displays make this design goal particularly interesting, and so it is no longer just a matter of awareness. Volumetric displays provide data in true 3D space, so it can be difficult to understand what another user is seeing, since when viewing 3D data, the viewpoint can impact how the data is perceived. Furthermore, because users cannot directly reach in and touch the data, it will be difficult to point to an area or indicate an area of interest without some form of virtual aid. As such, users need to not only be aware of what other users are doing, but also understand what other users are trying to communicate to them. We will explore techniques which will facilitate communication and awareness between users.

8.4 Exploration Platform

We developed an interactive 3D model viewing prototype to serve as a platform for our explorations. Users can view, inspect, label, markup, and modify 3D models in parallel. We chose this example application as it is general enough such that our interaction techniques could be applicable in a range of usage domains. Similar to previous prototype SDG applications, the application was developed to support two users [Agrawala et al. 1997, Wu and Balakrishnan 2003]. However, the majority of the interactions directly generalize to an arbitrary number of users.

8.5 Implementation Details

8.5.1 Display Device

We used the same 3D volumetric display that as used in the studies described in the previous chapters. A full description of the display is provided in Section 1.4.

8.5.2 User Tracking and Input

Users stood and were free to walk around the display. We used a Vicon motion tracking system (www.vicon.com) to track the positions of the viewers' heads. Three passive reflective markers were placed on a hat which users wore. Each user held their own 3D input device, consisting of a wireless presentation mouse, augmented with 3 Vicon markers (Figure 8-1). The devices had a left and right button which could be used for our interaction techniques.

Six Vicon cameras tracked the 3D location of the markers, and the data was streamed into our application at 120Hz. This hardware setup should be viewed simply as an enabling technology for our exploration, rather than one that would be used in any future real implementation.

Two keyboards were placed on opposite sides of the display which allowed for concurrent input from each user.

8.5.3 Software

The software was programmed in C++ using a modified OpenGL library for the volumetric display. The application ran on a Pentium 4 PC running at 2 GHz.

8.6 Interaction Techniques

The interaction techniques which we have implemented can be categorized as follows: interface controls, navigation, markup and manipulation, and advanced interactions.

8.6.1 Interface Controls

Users can change tools, execute commands, and change system options through a 3D radial menu and an options dialog box. In this section we will also described the two types of cursors which are used for interface control: the depth ray and the 3D point cursor.

8.6.1.1 3D Radial Menu

In the single-user application developed in the previous chapter, menus existed on the surface of the display, and were used through direct touch. This could cause an accessibility problem [Stewart et al. 1999] when there are multiple users, since there may not be a global location that all users could reach. Instead, similar to previous SDG applications, we use a popup menu [Shen et al. 2004, Wu and Balakrishnan 2003, Zanella and Greenberg 2001]. To use the menu, the user presses the right button of their input device. Users each have their own menu, so it supports *parallel access*. The menu is displayed in a visible location of the display volume, and is oriented towards the user who activated it, so it is also *location sensitive*.

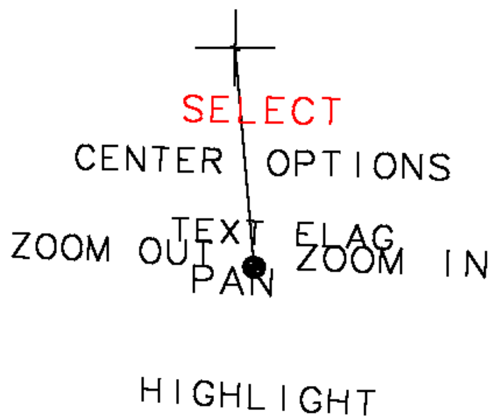

Figure 8-2. The 3D radial menu is used to switch between tools and execute commands

Since the display volume is 3D we felt it would be worthwhile to explore a *3D radial menu*. So, unlike traditional radial menus, the menu items, which consist of tools and commands, are distributed spherically in 3D space (Figure 8-2). A similar idea has been explored in 3D virtual environments [Grosjean et al. 2002]. Extending the menu to 3D increases the angle between menu items, which, in theory, reduces the motor constraints imposed on the user to select an item [Accot and Zhai 1997], and could potentially increase the efficiency of the menu.

Once the menu is activated, a 3D crosshair is displayed at its center, and the user controls it via a direct one-to-one mapping from the input device. The user moves towards the desired menu item and releases the button, or can release the button while still in the center area to cancel the menu.

8.6.1.2 Options Dialog Box

Where the 3D radial menu is used to switch between tools and execute commands, the *options dialog box* is used to control global parameters. It is activated through the 3D radial menu. The dialog box is a simplified version of a traditional GUI dialog box (Figure 8-3). Like the 3D radial menu, it is oriented towards the user, but it is displayed on a 2D plane. A standard cursor, controlled by the user's input device with a direct mapping, is projected onto the plane of the dialog box. The options dialog box is similar to the "pen & tablet" interaction metaphor used in immersive virtual reality environments [Bowman and Wingrave 2001].

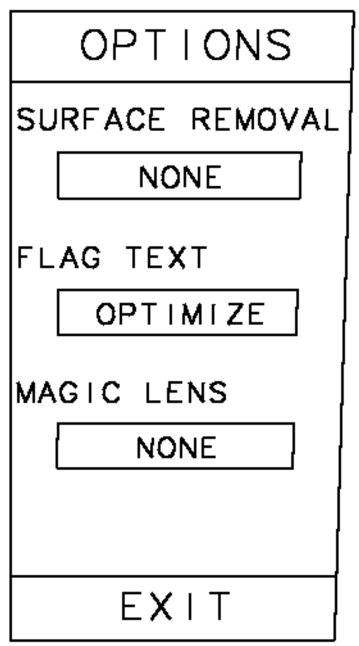

Figure 8-3. The options dialog box is used to control global parameters. The parameter values, displayed inside the rectangles, can be modified by clicking on them.

Users can click on various options to toggle values, and click the exit option to exit the dialog box. As with the 3D radial menu, users can access their dialog boxes in parallel. When both dialog boxes are active, changes in values on one dialog box are reflected immediately on the other dialog box.

8.6.1.3 Depth Ray

The *depth ray* is used as the selection tool for the system. The depth ray is based on the ray casting metaphor [Mine 1995a], and is fully described and evaluated in Chapter 6, where we found it to be an efficient selection mechanism for volumetric displays. It consists of a virtual ray emitted from the input device, and a depth marker, which can be moved forwards and backwards along the length of the ray. The depth marker is used to disambiguate when multiple selectable items are intersected by the ray. The intersected item closest to the depth marker is highlighted, and can be selected with a left click.

Each of the users can use their depth ray at the same time, so it supports *parallel access*. However an object highlighted by one depth ray is ignored by the other depth ray, which prevents both users from selecting the same object. Users can also use the depth ray to highlight objects without selecting them, to indicate a feature to the other user. Thus, the depth ray also supports *inter-user understanding*.

8.6.1.4 3D Cursor

While the depth ray is an efficient mechanism for selection, it can be awkward for defining 3D locations or for carrying out 3D manipulations. For tools that require 3D positioning, we use a *3D cursor* [Mine 1995a], which is more appropriate for such tasks [Bowman and Hodges 1997]. The cursor is rendered as either a sphere or a 3D crosshair, depending on the current tool being used.

For multiple users whose positions are not fixed, a strict absolute mapping, which we used in Chapters 5 and 6, will not work. To ensure that the 3D cursor, and tools which rely on it, would be omnidirectional, we looked at alternative mappings. A relative mapping would be difficult to implement, since there is no obvious clutching mechanism for a device which is held in midair. Instead, we define a dynamic absolute mapping which is relative to the user's location. As such, the 3D cursor supports *location sensitive interaction*.

A default vector V is used to define the offset between the user's location, L, and location where the input device would map the cursor to the origin of the display volume. The mapping uses V', which is V rotated by the users viewing angle, so that the offset is appropriate regardless of the users position around the display. The cursor is controlled with a control-display gain of 1. Thus, the cursor position, *C*, is defined as:

$$C = D - (L + V'),$$

where *D* is the position of the input device. To prevent slight movements of the user's head from changing the cursor location, we only update *L* when the cursor leaves the display volume. Thus, the absolute mapping is only updated when necessary, and the update is invisible to the user, since it happens when the cursor cannot be seen. Since each user may prefer to hold the input device in a different position, we allow users to manually set the vector *V*, using a simple calibration procedure. The user holds the input device in their preferred center location, and then presses and holds the right button, while keeping the device in the same location. After 3 seconds, and audio beep is heard, signifying an update in the default offset vector, to match the users preferred center location. This procedure would only need to be carried out once at the beginning of use.

8.6.2 Navigation

One of the benefits of viewing a virtual 3D model over a physical one is that users can easily inspect different areas of the model through navigation techniques that may not be possible in the physical world. Indeed, navigation has been identified as a "universal interaction task" for 3D environments [Bowman and Wingrave 2001]. Furthermore, navigation is particularly important for us to explore within a collaborative setting, because of the *coupled navigation* issue which could arise. To mitigate the effects of coupled navigation, as soon as one user begins any type of navigation operation, all other navigation operations are locked out, until the initial operation is completed. This does not eliminate potential problems, as one user may still navigate while the other user is in the middle of viewing something. By implementing the following navigation tools, we were able to explore such effects.

8.6.2.1 Location Aware Rotation

Since users can stand anywhere around the volumetric display, each user may have very different perspectives of the displayed imagery. Our rotation tool allows one user to see what another user is seeing, without having to physically walk around the display.

When users place their input device close the bottom portion of the display surface, a rotation widget fades in which indicates that a rotation can begin (Figure 8-4a). Clicking and holding the right button, while scrubbing either left or right, rotates the scene either left or right around the Z-axis. The rotation is spring loaded - if the user releases the button, the scene animates back to the original rotation. This allows users to take a quick glance of the model from another viewpoint, much like the "Glances" navigation tool [Pierce et al. 1999a]. Alternatively, users can pin the new rotation by sliding the input device upwards.

While rotating, viewpoint widgets are displayed for each user, indicating where their viewpoints were in relation to the scene before the rotation began. When a rotation begins these widgets are positioned directly between the user and the center of the display (Figure 8-4b), but the widgets rotate with the scene (Figure 8-4c). Users can see what another user's viewpoint was by continuing to rotate until that other user's viewpoint widget is aligned with their own viewing location (Figure 8-4d). The rotation will snap to this aligned location. Similarly, users can pass their viewpoint to the other user by rotating until their viewpoint widget is aligned and snaps to the other user's viewing location. This "viewpoint passing" interaction is *location sensitive* and supports our *inter-user understanding* design goal.

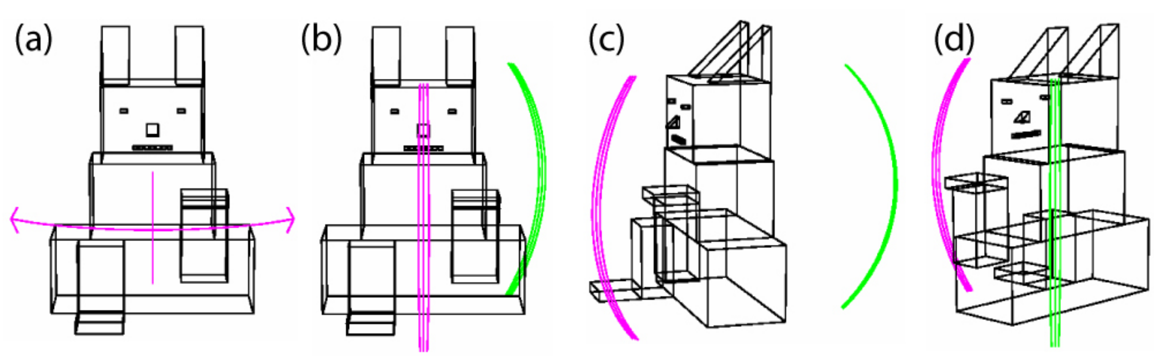

Figure 8-4. Location aware rotation. a) A widget is displayed indicating that a rotation can begin. b) When a rotation begins, viewpoint widgets indicate the user's own (pink) and collaborator's (green) viewpoints. c) Viewpoint widgets rotate with the scene. d) One user can align and snap to another user's viewpoint.[1]

---

[1] The figure used in this image was from a puzzle which tested a player's ability to match the 3D model with its associated 2D orthographic front view. This task was incorporated into our initial observation sessions, discussed in Section 8.7.

8.6.2.2 Panning

To pan the scene the user selects the panning tool. Once selected, a translation widget is rendered in the center of the display. Clicking the left button begins a pan, which is directly controlled by the position of the input device.

8.6.2.3 Zooming In

The *zoom-in tool* can be used to obtain a more detailed view of a certain area. The zoom-in tool utilizes the 3D cursor, which is displayed as a sphere (Figure 8-5a). A user can control the size of the sphere by twisting the input device left and right to decrease and increase the radius. The sphere acts as a preview to the new viewing volume; the smaller the sphere the higher the zoom level. Clicking the left button animates the scene to the new zoom level (Figure 8-5b). This is similar to the traditional marquee zooming, however, the contents of the sphere indicate to the user what the new view will be before committing to the zoom. The other user can also see this preview, with the opportunity to comment. Thus the zoom-in tool supports *inter-user understanding.*

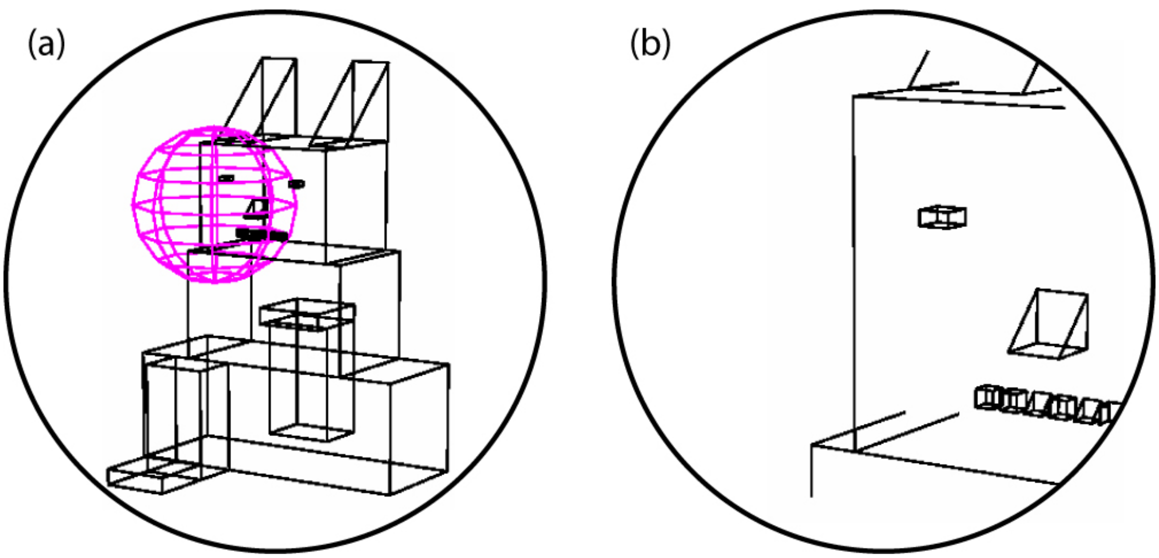

Figure 8-5. Zoom-In Tool. a) The user controls the position and radius of a spherical 3D cursor. b) Clicking the left button zooms in to the area represented by the sphere. Circles represent the display volume.

8.6.2.4 Zooming Out

To zoom out the user selects the *zoom-out tool* from the radial menu. When selected, a zooming widget is displayed in the center of the viewing volume. The user can hold the left button down and pull away from the display surface to decreases the zoom level. The zoom is always centered at the origin of the coordinate system, such that once the default minimum zoom level is reached, the model will be centered inside the viewing volume.

8.6.3 Markup and Manipulation

Regardless of the usage domain, users will likely want some way to highlight, and possibly even modify, areas of the displayed data. The ability to markup data is important for *inter-user understanding*, since it allows users to indicate areas of interest to one another. We explored the following markup and manipulation tools.

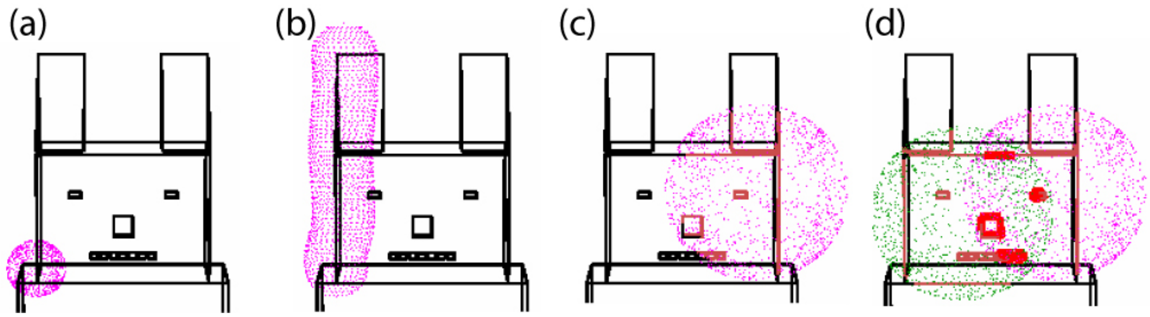

Figure 8-6. Highlighting tool. a) The highlighting tool is displayed as a spherical cloud of points. b) Holding down the left button adds a new highlight cloud. c) The highlighting tool can be used as a magic lens. d) The magic lens effect is amplified inside the intersection of both users' tools.

8.6.3.1 Highlighting Tool

Since users cannot reach into the virtual image, it can be particularly difficult for one user to indicate an area of interest to another user. The *highlighting tool* allows users to highlight 3D areas, so that they can explicitly define areas of interest to other users. Thus, the tool supports *inter-user understanding.* The tool is an extension to the 3D annotation tool developed by Tsang et al. [2002], which only allowed 2D annotations projected onto 3D geometry. The highlighting tool is controlled by the 3D cursor, so it is *location sensitive.* It is rendered as a spherical cloud of points (Figure 8-6a). Pressing the left button adds a "highlight cloud" to the scene. The user can move the cursor while the left button is pressed to sweep out a freeform 3D area to highlight (Figure 8-6b). Only exterior points of the cloud are displayed, so that the highlighted region consists of an outer surface of points. Like the zoom-in tool, the radius of the cursor can be modified by twisting the input device, allowing users to sweep out coarse or detailed regions.

This tool can support *inter-user understanding* without adding a highlight cloud. A user could use the cursor to indicate an area of interest to another user. To aid in this usage scenario, we provide volumetric magic lens [Viega et al. 1996] functionality to the highlighting tool cursor, which is activated through the options dialog box. When active, all elements of the scene within the bounds of the spherical cursor appear as a different color (Figure 8-6c). This magic lens functionality can also be used collaboratively - if both users have their cursor in the same area, the elements of the model within the intersection of the two cursors appear much brighter (Figure 8-6d). While the composition of two volumetric lenses has been previously explored [Viega et al. 1996], our implementation is in a true 3D volume, and allows separate users to each control a lens in parallel. The magic lens can also be set to cull out anything within its bounds

The highlighting tool supports *parallel access* as both users can create highlight clouds in parallel without any constraints. The color of any created highlight cloud matches the cursor color of the user who created it.

8.6.3.2 Text Flags

In addition to marking up a model with the highlighting tool, users may also want to label certain areas of interest. While 3D annotations have been previously explored [Tsang et al. 2002], new issues arise in a collaborative volumetric display scenario. Specifically, the label can be positioned in true 3D space, and the label must be readable by all users.

To add a label, the user selects the *text flag tool* from the 3D radial menu. The text flag tool uses the 3D cursor, which is rendered as a crosshair. The cursor is used to create text flags, which consist of a line with a rectangle containing text at one end (Figure 8-7). The other end of the line is considered the origin, which is the 3D location being labeled. If the user wishes to label a general area, rather than a specific location, then the text flag tool could be used in combination with the highlighting tool.

To create the flag the user positions the cursor at the 3D location of an area which they wish to label. Clicking the left button sets the origin of the flag. While the left button is still down, the user can position the other end of the flag. The user can then type on their keyboard to add the text.

As we discussed in Chapter 4, a challenge with presenting text on the volumetric display, when multiple users are present, is orienting it so that it is readable. We provide three possible modes of display for the text flags which address this orientation issue. This mode is set in the options dialog box.

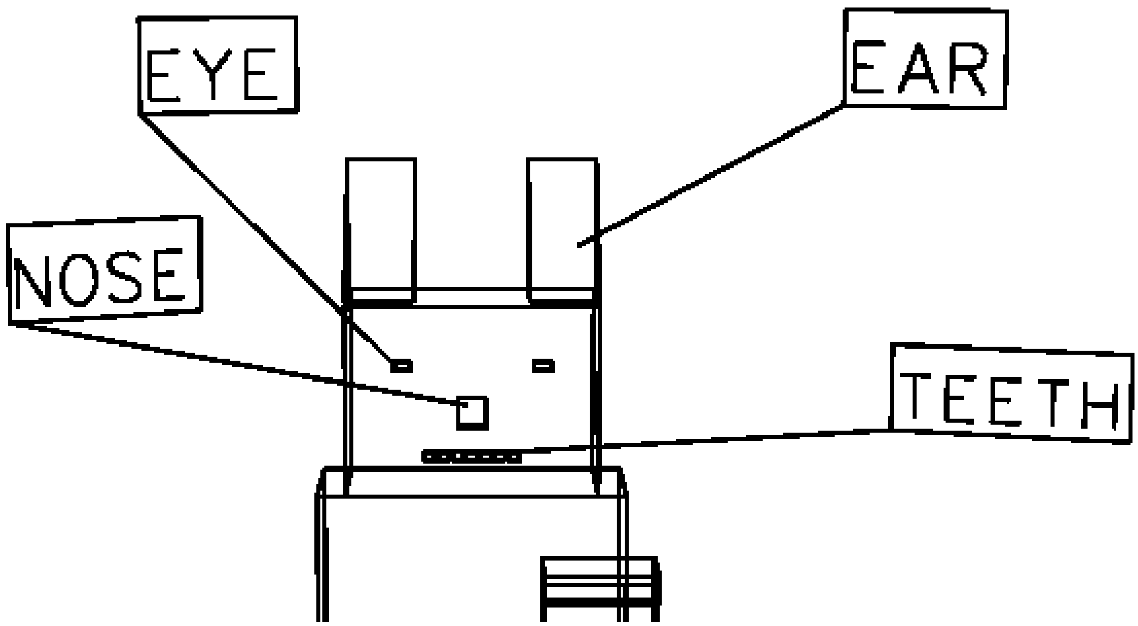

Figure 8-7. Text flags can be used to label areas of a model.

The "optimized mode" optimizes the orientation of the flag to minimize the reading time for *both* users, so it supports *parallel access*. This is a direct implementation of the optimization algorithm described in Section 4.5. The algorithm is based on the real-time location of the users, and so it is *location sensitive*. In the cases where the text will need to be forwards to one user and backwards to the other, preference is give the user who created the text flag and it will appear forwards to that user. Users can override this algorithm by placing their cursor close to the text flag, causing it to temporarily orient towards that user.

The "rotate" mode causes all text flags to slowly rotate about the Z-axis. This gives each user a chance to see the text from the optimal orientation.

The last mode is a "privacy" mode. With this mode, the rotation is set such that the text is hidden from the user that did not create it, by keeping the text parallel to the other user's line of sight. Such private viewing has been previously suggested for SDG applications [Agrawala et al. 1997, Wu and Balakrishnan 2003]. This mode works best if the two users are standing at 90° from each other, since the text will be facing the user who created it, and parallel to the other user. The mode does not work well if the users are at exact opposite sides, since the text will be parallel and thus hidden from both users.

We implemented a simple layout algorithm to update the positions of the text flags during use of the navigation tools. The algorithm guarantees that if the origin of a flag is in the viewing volume, then the text portion of the flag will also be visible. If the origin of the text flag is not in the viewing volume, then the text flag is not displayed.

As with the highlighting tool, the text flag tool supports *parallel access* as it can be used in parallel by both users. Text flags, which are not private and hidden from a user, can be selected and deleted by the depth ray, via a contextual popup menu. Once selected, the text within the flag can be edited, and the location of its endpoints can be modified. When a user selects a text flag, the text orients towards that user, regardless of the viewing mode.

8.6.3.3 Object Manipulation

While our prototype application was built around the scenario of *viewing* a 3D scene, there may be cases where users wish to *manipulate* the data. For example, when a team of car designers are viewing a car, one designer may want to demonstrate an idea to slightly change the curvature of the front hood. The availability of a manipulation tool could thus support *inter-user understanding*.

Manipulations are carried out with the depth ray, which can be used to select faces of the model (Figure 8-8a). Once selected, the user can drag the face along its normal vector by moving the input device (Figure 8-8b). More elegant modification tools can be imagined, but are beyond the scope of our work. Along with supporting *inter-user understanding,* these manipulations supports *parallel access,* as users can both modify the scene in parallel.

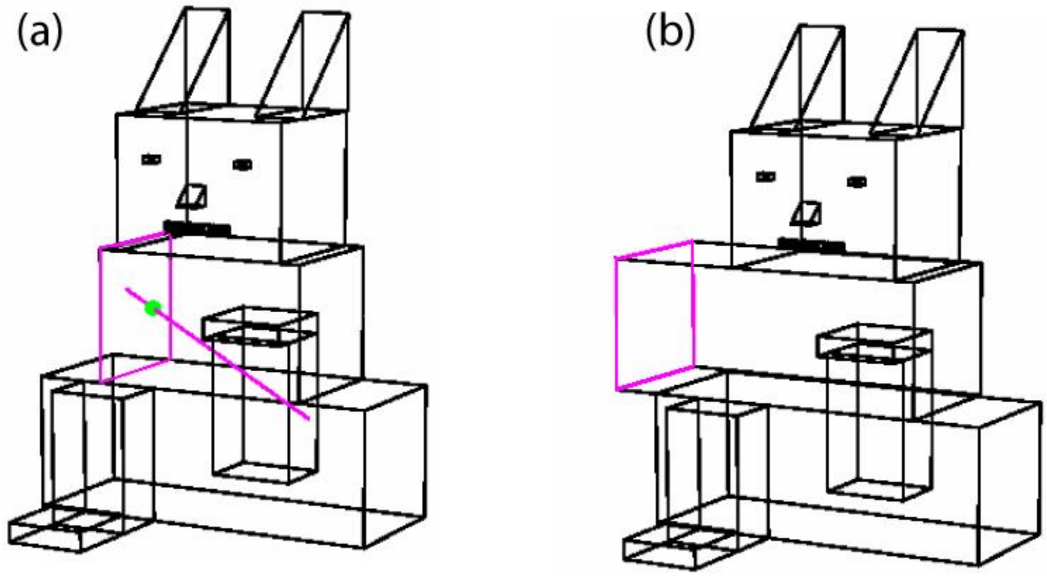

Figure 8-8. The depth ray (a) can select and drag faces (b).

8.6.4 Advanced Features

8.6.4.1 Scene Splitting

We have described a number of operations, such as highlighting, labeling, and manipulating, which support *parallel access.* However, if users wish to work on areas of a scene which are too far apart, or require different zoom levels, then they may not be able to carry out the tasks in parallel. To support *parallel access* under such scenarios, we implemented *scene splitting.*

Scene splitting divides the viewing volume into multiple viewports. This idea has been previously suggested but never implemented within volumetric displays [Balakrishnan et al. 2001], although the concept has been explored in collaborative synchronous text editors [Baecker et al. 1994]. Users split the scene with a slicing gesture across the surface of the display with their input device. A 2D dividing plane rendered as a grid is drawn across the display volume, which divides the volume into the two viewports, one for each user (Figure 8-9).

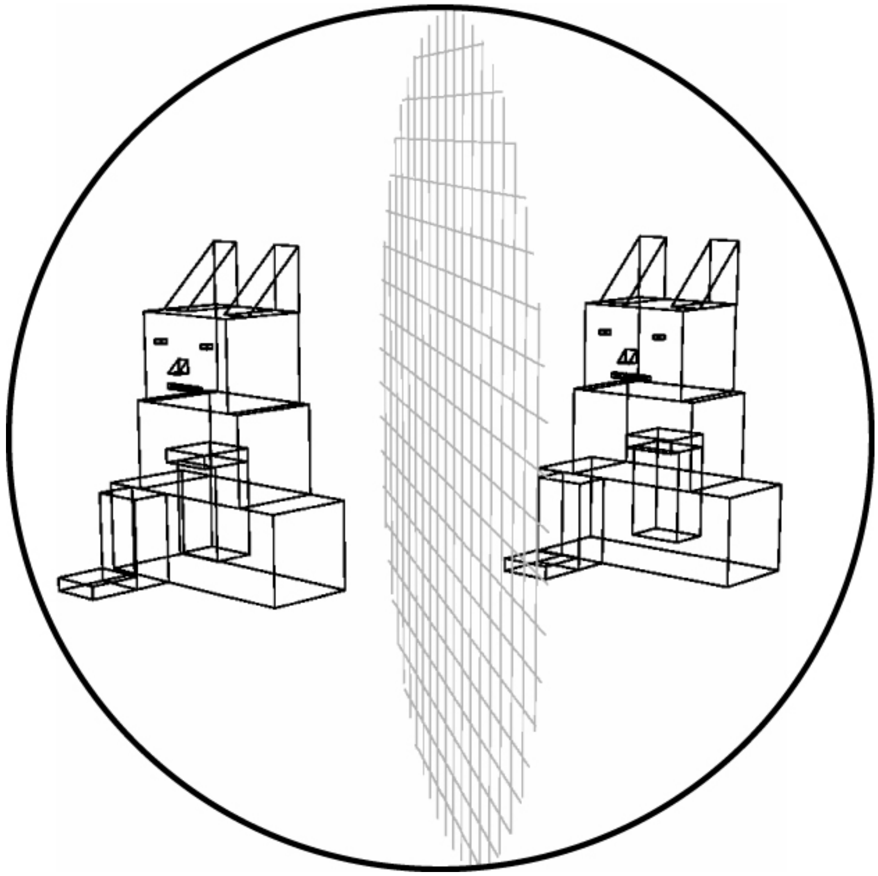

Figure 8-9. When the scene is split a grid divides the display space into two viewports, one for each user. The circle represents the display volume.

Once the scene is split, users can carry out all of the previously described operations on their side of the display, with no effect on the other user's viewport. However, if one user is zoomed in, the location which they are working is indicated to the other user by displaying a sphere on the other's scene with the appropriate location and size. This feature supports *inter-user understanding*. To minimize distraction, text flags and highlight clouds which are added by one user do not appear in the other user's scene. However, any face manipulations are reflected immediately in the other user's scene to eliminate the need of conflict management techniques [Greenberg and Marwood 1994].

To return to a single viewport, the scenes can be merged by dragging the input device along the surface of the display from one side of the dividing plane to the other. When the scene is merged all text flags and highlight clouds that were added by either user while the scene was split are shown.

8.6.4.2 Hidden Surface Removal

As we have already discussed, one general limitation of almost all volumetric displays is that they are incapable of exhibiting occlusion of one part of the image volume by another (Figure 8-10a). This is because the light which illuminates a voxel is omnidirectional [Favalora 2005]. However, if an application is *location sensitive*, then surfaces which are behind other objects based on the user's viewpoint can be manually hidden. Unfortunately, this can only be done for a single user - since a surface which is hidden from one user may be visible to another. To compensate for this, we support three modes of hidden surface removal, which are set from the options dialog box.

The first two modes correspond to activating hidden surface removal relative to either the first or second user's viewpoint. Our algorithm clips all lines which are not visible based on the location of the midpoint between the user's eyes (Figure 8-10b). The location of the eyes is estimated using a default vector from the location of the user's hat. If more precision is desired, the exact offset between the eyes and the user's hat can be determined using a short calibration program. While these two modes will hide surfaces which the other user should be seeing, it could still be useful if the users want to take turns to see a more realistic view of the scene.

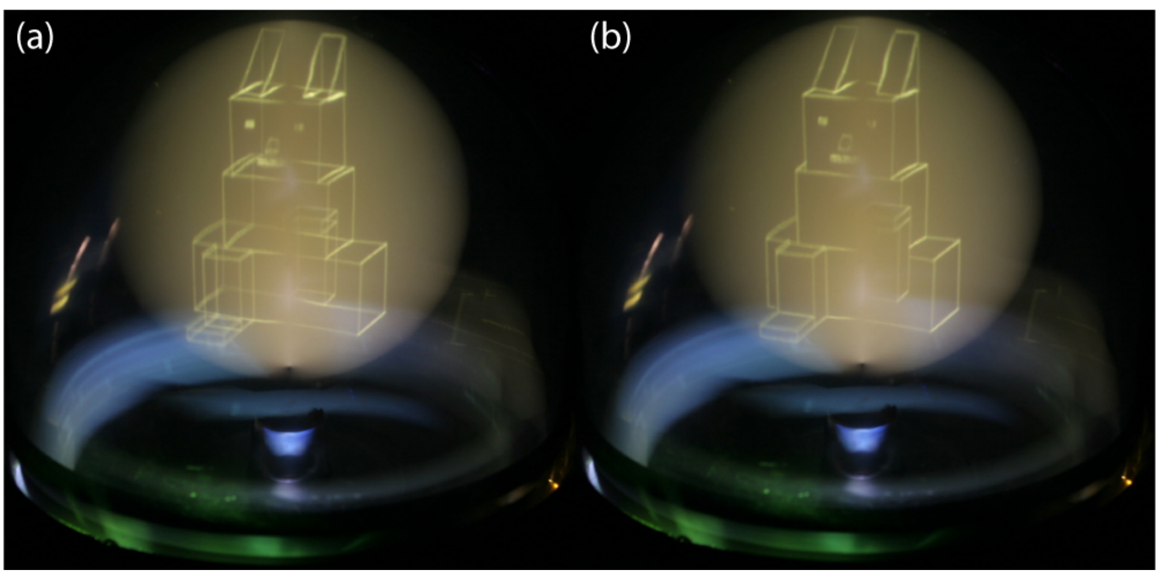

Figure 8-10. a) Typically volumetric displays are incapable of hidden surface removal. b) The same scene displayed using our hidden surface removal algorithm, which is updated in real-time based on the user's viewpoint. For illustration in this figure, the algorithm is based on the location of the camera.

The third option is a merged hidden surface removal rendering of the scene. This option will render the union of what both users can see from their viewpoints. The effectiveness of this mode depends on the relative location of the users. Alternatively if both users wanted to see the scene with accurate hidden surface removal, they could split the scene to get their own views.

8.7 Initial User Observation Sessions

We conducted three observation sessions, each with two different `volunteer users working with the prototype application we have described in this chapter. A think-aloud protocol was used, and sessions lasted approximately one hour. Each session began with a 15 minute instruction period, where participants tried out all of the features of the system. After the instruction period, an abstract model which resembled an animal figure (Figure 8-10) was displayed and the two users were asked to carry out 4 tasks, of approximately 5-10 minutes each.

In the first task users worked together to label 10 features of the model. In the second task each user was given a different list of features to label, requiring *parallel access.* In the third task one user indicated areas of interest for the other user to label, requiring *inter-user understanding.* The fourth task was meant to evaluate the various interaction techniques in combination: users were given a diagram which had 10 differences from the displayed 3D model. Users were asked to identify these 10 differences, and to then correct the differences on the virtual model.

8.7.1 Observations

The users were able to use most features of the system without difficulty after short instruction, and users were able to complete all tasks. The final task, in particular, went well, as participants successfully used a combination of the rotation, navigation, highlighting, and text flag tools to complete the task. The users made comments about specifically liking the optimized text flag rotation, the ability to manipulate models in parallel, and the hidden surface removal. More detailed observations in terms of our design goals, and encountered difficulties, are now discussed.

8.7.1.1 Location Sensitive Interaction

An interesting overall observation was that the groups used a combination of both the virtual rotation functionality, and "physical" rotation (walking around the display). In two of the three groups, users walked right around one another. The fact that users were able to use the techniques and complete the tasks while physically walking around the display indicates that our techniques successful supported *location sensitive interaction*. All of the users seemed to find the dynamic absolute 3D cursor mapping easy to work with, and most of the users liked how the orientation of the menus and text flags were updated based on their locations.

8.7.1.2 Parallel Access

Overall, the users worked well in parallel. This was especially true in the second task, where users worked completely independent of one another. The interface elements for each user did not interfere with one another, except for a few instances when both users had their menus up at the same time. Two of the users did comment that they were sometimes reluctant to perform virtual navigations as they did not want to change the viewpoint when the other user was doing something. With all groups, users often verbally discussed navigations before executing them. For example "let's zoom in again", or "we'll do the teeth last, because we have to zoom in".

8.7.1.3 Inter-User Understanding

The users took advantage of the tools which were provided to support *inter-user understanding*. In some cases they explicitly switched to the highlighting tool to point out areas to each other, but at other times they just used the location of their current tool cursors, to avoid an explicit tool change. One participant used the depth ray to indicate faces to the other user, because the depth ray could highlight an entire face. During the third task in one of the sessions, one user would highlight an area, pass his view to the other user, and then the other user would label it. The fact that the tasks were completed successfully indicates that our techniques did support communication and awareness between users.

8.7.1.4 Interface Limitations and Difficulties

Half of the users found it hard to use the zoom-in tool. This was in part due to difficulties in twisting to change the zoom level, but was also because users were used to a marquee zooming tool. Furthermore, two of the users did not like having separate tools for zooming in and out. In response to this, we iteratively redesigned the zoom-out tool such that the user could position the 3D cursor to set the center of the zoom, and then click the left button and move the device forwards or backwards to zoom in our out.

Two of the users also did not like how for some tools, such as panning, no cursor was drawn, which caused confusion about which mode they were in. We updated the application such that the 3D cursor is displayed for such modes, even if it does not serve any purpose other than revealing system status.

One last limitation of our system was that the keyboards were not always accessible when users wished to add a text label. In one group, the two users had switched positions and a user tried using the keyboard which was in front of him, which didn't work, since the keyboards could only send input to their "owner's" text flags. To address this, multiple keyboards, with the ability to distinguish input between users, could be placed around the display.

8.8 Domain Expert Interviews

In addition to the observation sessions, we conducted interviews with experts in potential usage domains. We met with three anatomy professors and one professional landscape architect. The interviews lasted about 60-90 minutes. The entire system was demonstrated and feedback was obtained throughout the interview.

The reactions in each of our interviews were quite positive. The overall response from all of the interviews was that the system had "tremendous potential". The anatomy experts said the system would be great for education, diagnosis, and in particular, surgical planning, due to the number of elements involved with complex 3D relationships. The architect said the system had a "huge range of prospects", with "almost no end of encouraging future applications", and that it would be great to incorporate in the design process for understanding relationships.

Another high-level response was the appreciation for a truly 3D virtual platform which could replace or at least complement current physical processes. One anatomy professor discussed how “anatomy museums”, which display physical specimens inside enclosures for educational purposes, are burdened by security and storage costs, leaking fluids, plus legislative issues, since they involve biological specimens. A volumetric display would not suffer from these drawbacks, while maintaining the affordances of an enclosed physical 3D specimen. The architect commented that scaled-down physical models provide a necessary 3D viewing modality, but often prohibit creative design, because any mistake is costly, with respect to time and the cost of materials. If the model was instead displayed virtually inside a volumetric display, then the designer could worry less about making specific mistakes, and concentrate more on the creative design, while still obtaining the desired 3D viewing mode. We also obtained specific feature comments:

- Anatomy experts liked the rotation, as it replicated the physical Lazy Susan’s used in anatomy museums.
- The architect liked the ability to physically walk around the display, and said it would allow designers to use their “innate biological resources” to understand the spatial relationships of a model. This comment is validation for our *location-sensitive interaction* design goal.
- The architect liked the mark-up tools and said it would allow designers to “analyze and interrogate” a 3D model.
- An anatomy expert appreciated the highlighting tool, and mentioned that it is often difficult to point out a feature to a student when a specimen is in an enclosure.
- Both the architect and anatomy experts said the culling functionality of the highlighting tool would be great to reveal inner relationships during a virtual 3D dissection.
- One anatomy expert liked that text labels could be added, moved, and deleted, because it would allow the labeling of a model to change over time. In a physically encased model, the inner labels are static and cannot be changed.
- The architect liked the idea of supporting subtle manipulations, as it would allow a designer to edit a model during the demonstrations which are typically done with rigid physical 3D models.

Finally, a number of new features were suggested. An anatomy professor wanted to see support for remote collaboration, so a surgeon in a different country could help plan a procedure, or a remote group of students could attend a virtual dissection. The ability to record usage sessions for future playback was also suggested. The architect mentioned that it would be interesting if the system was much larger, and provided a more immersive viewing angle of the scene.

In summary, the experts all felt the system had great potential, saw numerous potential applications, and were impressed by a number of the features which were implemented. The fact that so many possible usage scenarios were suggested informs us that we were successful in developing interaction techniques which could be applicable to various usage domains.

8.9 Discussion

Although our prototype was developed for two users, in general the techniques and application which we implemented could be used by an arbitrary number of users without modification. An exception is the scene splitting functionality, which would require new considerations. A user may want a split scene which is shared by some, but not all of the other users. The system could also support the splitting of a viewport which was already split but shared. One feature that would not be possible with more than two users is the privacy viewing mode for text flags, since a text flag could only be hidden from one user.

Our location-aware assumption requirement was more critical to our application. While the Vicon markers gave us extremely accurate locations for the users, most of the system functionality would still work fine if precise positions were unknown. Alternative non-intrusive technologies which would provide a lower grade estimate of the user's location could be implemented. For example, stereo computer vision, sonar, or a pressure activated floor mat could all provide approximate body locations and would be invisible to the user. In the absence of these technologies, the input device locations could be used to infer the user locations. The only feature of our system which would significantly degrade with such less accurate position information is the hidden surface removal.

More techniques would be affected if the system had no viewer location information at all. The viewpoint passing would not be possible, and neither would the optimized text flag viewing mode. A new cursor mapping would also be required. One possibility would be to use a relative mapping from the input device location, and to disallow the cursor from leaving the display volume.

8.10 Implications for User Interface Design

The work presented in this chapter has direct implications for the future design of collaborative applications for volumetric displays. We intentionally chose a generic model-viewing application to serve as a platform for our explorations, so that the interactions which we developed could be implemented in various applications, regardless of the usage domain. In addition to contributing these new techniques, our design principles can serve as a baseline for designing of new collaborative user interfaces and interaction techniques. The positive results from our user observations confirm that our techniques could be integrated into future applications.

8.11 Future Work

Since our work is an initial prototype, which implements a number of new techniques, there are a number of lines of future work which could be followed.

Firstly, our system could be evaluated with a formal user study, where users perform a real world 3D task which is currently carried out with other technological platforms or even in the physical realm, in a specific usage domain, such as medical education. The study could investigate the difference which arise when the task is completed using the volumetric display. Furthermore, a number of the new techniques which were implemented could also be studied in isolation. For example, an empirical evaluation could be performed on the 3D marking menus, to test if the extra dimensions allows for more items to be placed on the menu without loss of efficiency. The effectives of both the individual and merged hidden surface removal viewing modes could also be evaluated.

Along with empirical evaluations, the design of a number of the techniques which we presented could be extended. For example, the merged hidden surface removal algorithm could be implemented such that objects which could only be seen by one user appear dimmer than objects which were visible to both users. Alternate cursor mappings, such as a relative mapping with clutching could also be investigated. It would also be interesting to explore the highlighting tool as a mechanism for selecting multiple objects.

Finally, based on comments obtained from expert interviews, a new line of work which could be conducted is studying our system as a tool for remote collaboration. If the technology was available at two different locations, users could simultaneously view and possibly even interact with a usage session occurring at a remote location. Alternatively, the system could record usage sessions, and remote users could play those sessions back at a later time.

8.12 Conclusions

As we have discussed in our introduction to this chapter, volumetric displays possess a number of unique properties which make them particular suitable for collaboration. Users do not have to wear supplementary hardware, and a get a 360° viewing angle.

In our work in Chapter 7, we explored user interfaces for volumetric displays, but only considered a single-user application. In this chapter we developed on that work and explored the issues surrounding collaborative interaction with volumetric displays. We discussed and implemented a number of new interaction techniques, adhering to three design goals. We also performed usage observations of participants performing a series of tasks, and interviewed a number of domain experts to obtain feedback on our system.

The results of these observations and interviews were very encouraging. Users were able to complete all of the tasks, and had a number of positive comments about the system. The expert interviews confirmed our belief that volumetric displays hold a great potential for collaborative applications, as a great deal of enthusiasm was expressed about the system. The encouraging observations made during the usage sessions, in combination with the positive feedback received during our expert interviews, indicate that our work can serve as a guideline for the future development of collaborative applications for volumetric displays.

9. Conclusions

"The future ain't what it used to be."

- Yogi Berra

9.1 Summary

Researchers have been developing 3D displays in an effort to provide more realistic imagery for more than 50 years. These displays leverage the physiological mechanisms which a human uses to perceive depth. They range from non-immersive stereoscopic displays using shutter glasses, to fully immersive head mounted displays. Such displays have the potential to improve a user's ability to carry out virtual 3D tasks. Unfortunately they possess a number of drawbacks which have prevented their widespread adoption. Generally, they require the user to wear some sort of device, whether it is shutter glasses, or a head tracker. Furthermore, the accommodation and convergence cues which they provide are inconsistent, because the actual location of the imagery is on a fixed display plane. This discrepancy can cause headache, fatigue, and nausea among users.

Volumetric displays, which are a more recent class of display to emerge, are free of these critical drawbacks. However, because they have only recently become available for use, there is little understanding about the human factors and interaction issues associated with their use. It is important to obtain a sound understanding of these issues, before interactive applications for volumetric displays are developed. This thesis is a first step towards investigating these issues. We first identified the unique properties of volumetric displays, and then examined the effect of these unique properties on low-level human factor issues through theoretical models and formal empirical experiments. Based on the understanding obtained from these low-level investigations, we contributed new interaction techniques for single and multiple user applications, and evaluated these designs through formal experimentations, user observations, and expert interviews.

In Chapter 1 of this thesis, we provided a discussion of the unique properties of volumetric displays, and discuss why these properties raise new interaction questions which are not prominent and have not been answered in association with other 3D display forms. The properties which we discussed are the autostereoscopic display of imagery, the true 3D imagery, the 360° viewing angle, and the physical enclosure. Discussing these unique properties allowed us to identify the important human factor questions and interaction issues to explore, and motivated our investigation into these issues in the remaining chapters of this thesis.

Before initiating this investigation, we first provided a detailed review of background literature in Chapter 2. We discussed relevant 3D display research, devices for 3D input, 3D interaction techniques, 3D user interfaces, and collaborative applications. While a significant amount of this work was used to guide and inspire are own work, the unique properties of volumetric displays required a number of these important issues to be revisited.

The first of these issues which we addressed, in Chapter 3, is the human's ability to perceive depth in the true 3D display volume which volumetric displays provide. This allowed us to compare an important perceptual quality of the display to other forms of 3D displays, allowing us to understand if and when the display would be potentially beneficial for the completion of virtual 3D tasks. While depth perception is only one of many ways to assess the viewing quality of the display, we identified it as the inherent factor which volumetric displays potentially improve, since the displays provide consistent convergence and accommodation cues. Furthermore, understanding depth perception in volumetric displays could also serve as a base for other perceptual issues to be modeled and assessed. The results showed the volumetric displays did improve the perception of depth in comparison to a high-quality stereoscopic display with head-tracking. However our study also revealed that the quality issues associated with the display need to be addressed before the display can be deployed and used for real-world applications.

Continuing our investigation into perceptual issues associated with volumetric displays, in Chapter 4 we turned our attention to a user's ability to read text under 3D rotations in volumetric displays. As with the study in Chapter 3, this study was motivated by the unique properties of volumetric displays, and in this case, the display's 360° viewing angle. This property allows multiple users to stand around the display and view its imagery form different viewpoints. As a result, textual labels may be read from various angles. Our goal in Chapter 4 was to identify the effects of reading such text, and to consider techniques for mitigating any effects. In an initial experiment we found that text which undergoes a rotation of less than 60° can be easily read, rotations about the horizontal axis are most problematic, text which is parallel to the users' viewpoint is very difficult to read, and a large source of problems stems from the ambiguity of characters under rotations. Using the observations and empirical data obtained we designed and implemented an orientation optimization algorithm, which minimizes reading times for a group reading a textual label form different viewpoints. This technique was evaluated in a follow-up experiment, and was found to decrease average group reading times by 33%.

In Chapter 5 we shifted our attention to the human factors associated with motor control when interacting in a true 3D volume. We investigated pointing at targets in 3D, since pointing is a necessary task for many interface operations, including selection, which is at the core of almost every user interface. Obtaining an understanding of pointing in 3D was important, to allow us to develop appropriate interaction techniques for volumetric displays, and would also serve as a base for understanding and modeling other higher level operations, such as docking and tracking. We proposed a new predictive model for 3D pointing based on Fitts' Law, and validated this model in a formal experiment. The analysis of the experiment also allowed us to identify some important issues associated with pointing in 3D, and to make recommendations about user interface and interaction technique designs.

The results from this study were directly applied in Chapter 6, in the design and evaluation of new selection techniques for volumetric displays. In an initial experiment we demonstrated that the ray cursor is a more efficient selection technique than a 3D point cursor. Motivated by this result, we designed a number of new techniques to address the ambiguity problem associated with the ray cursor. In a second experiment, we implemented an evaluated these designs, and found the depth ray to be the most suitable candidate.

In Chapters 7 and 8 we continued our exploration into interaction techniques for volumetric displays. However, unlike selection techniques, which were explored in isolation, we explored remaining interaction techniques, such as menus, object manipulations, and navigation, within working user interfaces. This decision was made because unlike selection, the design of higher-level techniques has more dependence on the properties of the user interface, and the quality of such techniques is more subjective in nature, and less prone to experimental evaluations. In Chapter 7 we explored a user interface for volumetric displays using the display enclosure as the main input device for all interactions. We implemented important user interface elements and operations, such as a file browser, menus, and object transformations. In Chapter 8 we focused our exploration on collaborative interaction techniques. We revisited the important issues which arise in the development of single display groupware applications [Stewart et al. 1999], focusing on issues with particular relevance to volumetric displays. The techniques were developed within a working prototype that allowed two users to simultaneously view, markup, and manipulate 3D models. The system was evaluated through informal user testing, and feedback from potential domain experts was obtained through series of interviews. The results of these evaluations indicated that our developed techniques could serve as a baseline for future collaborative applications for volumetric displays.

9.2 Research Contributions

The thesis which we have presented covers a range of interesting areas related to the understanding of interaction issues associated with volumetric displays. Here we provide a summary of the major and minor research contributions of the work:

9.2.1 Major Contributions

- An empirical finding showing that volumetric displays have the potential to improve a user's ability to perceive depth, in comparison to more traditional 3D stereoscopic displays (Section 3.3).

- An empirical study showing that our new text orientation optimization algorithm significantly improves reading times when multiple users attempt to read a single textual label from different viewpoints (Section 4.6).
- The development of a number of new 3D Fitts' Law models which can predict movement times when pointing at targets in 3D space (Section 5.3.3), along with an empirical comparison and validation of these models, showing our weighted Euclidean model to perform particularly well (Section 5.4).
- The development of four new ray casting selection techniques which address the ambiguity problem associated with traditional ray cursors (Section 6.4).
- The implementation of an interactive 3D model building application for volumetric displays which explores a number of user interface components for volumetric displays, and investigates an interaction paradigm where the user interacts directly with the display by performing various multifinger gestures on and above the display surface (Sections 7.4 - 7.6).
- The implementation of a prototype collaborative 3D model viewing application used to explore the interaction issues raised when multiple users interact with the display (Sections 8.4 - 8.6).

9.2.2 Minor Contributions

- The identification of unique affordances possessed by volumetric displays, which should be considered when developing interactive applications (Section 1.2).
- The identification of ambiguity as a primary cause for difficulty when reading textual labels under suboptimal orientations in the volumetric display, and two proposed techniques, underline and uppercase rendering, to alleviate this ambiguity (Section 4.3).
- An empirical study which identified the effect of text orientation on reading times, and showed the uppercase rendering to be particularly beneficial (Section 4.4).
- The development of an orientation optimization algorithm for textual labels which are being viewed by multiple users (Section 4.5).
- An empirical investigation of the effects of the relevant factors in a 3D pointing task (Section 5.4).

- An empirical comparison showing that a ray casting metaphor outperforms the point cursor selection technique in a volumetric display selection task (Section 6.3).
- An empirical study comparing our newly development ray cursor selection techniques, showing the depth ray to perform particularly well (Section 6.5).

9.3 Considerations for Displays Having Other Properties

The investigations which we conducted, as summarized above, were based on the defining properties of volumetric displays, which we discussed in Section 1.2. These properties hold for all current forms of volumetric displays which we are aware of. However, it is conceivable that future implementations of volumetric displays may be characterized by others sets of properties. As such, it is worthwhile to consider how our work would generalize if these properties were changed. Here we revisit each of the properties discussed in Section 1.2, and consider the impact on our work if those properties were no longer upheld.

9.3.1 Autostereoscopic Display of Imagery

Since the current implementations of volumetric displays do not require users to wear supplementary hardware, we do not expect future volumetric displays to impose this requirement. Furthermore, this property has little impact on the interaction techniques which we developed, and the human factor studies which we carried out. If the supplementary hardware somehow changed the depth cues received by the user, it would probably be worthwhile to repeat the depth perception studies which we carried out in Chapter 3, since the results could be affected. Furthermore, depending on the intrusiveness of the hardware, the collaborative interaction scenario, explored in Chapter 8, may become less feasible, since users could lose the context of their surrounding collaborators.

One possible use of such hardware could be for head tracking, so that the system was aware of its users' locations. However, we have already considered a system which had this information, in Chapter 8. As we discussed in that chapter, it is our hope that such systems could obtain head tracking without requiring users to wear anything, through less intrusive techniques such as stereo computer vision, sonar, or a pressure activated floor mat.

Another reason to require users to wear hardware could be to provide user specific views of the displayed imagery, such as those explored by Agrawala et al. [1997]. If the users wore shutter glasses, then the display could be synchronized to those glasses, so each user saw their own view. This could allow for viewpoint dependent effects, such as surface reflection and occlusions, which are not currently possible. However, the rotation speed and projector frame rate would need to increase by a magnitude equal to the number of users, to maintain the same frame rate for each user.

9.3.2 Consistent Depth Information

By definition, volumetric displays present imagery in true 3D space. As such, similar to the above property, it is hard to imagine how a volumetric display would not provide consistent depth information. One possibility is if we relaxed definition of what a volumetric display is. For example, a volumetric display could be defined as a display which provides a *perceived* 3D display volume inside of a physical enclosure. One example could be a spherical display, where the imagery is projected onto the surface of the sphere, and not in the interior of the sphere [Lang et al. 2003]. With stereo glasses it could produce an effect of a volumetric displays space. Another example would be to have a single display plane inside of a hemispheric dome, which was dynamically oriented so that it was always facing the user, like an automatic chameleon display [Buxton and Fitzmaurice 1998]. Again, with stereo glasses, the imagery could be perceived as existing in true 3D space.

While such examples could potentially improve the imagery quality, users would no longer receive consistent depth cues, so our study on depth perception in Chapter 3 would no longer apply. Other than this study, as long as the other affordances remained the same, and the perceived display space remained the same, the interaction techniques which we developed could still be utilized.

9.3.3 Full 360° viewing

It is conceivable that future implementations of volumetric displays will have restricted viewing angles because of their technological implementations. If this were the case, the implications of the work in this thesis would need to be reconsidered. If the display could only be viewed from one side, then the orientation optimization algorithm for text, presented in Chapter 4, would be less necessary, as users' viewing angles would be less divergent. Furthermore, the collaborative scenario would become less appealing, as the users would need to be crowded within the possible viewing angles. However, as for the interaction techniques which we developed, none relied on a 360° viewing angle. In Chapter 7 we considered multi-viewpoint, and out-of-viewpoint interaction, however the techniques which were developed did not require the user to be able to walk right around the displays. Multi-viewpoint interactions support the users' ability to walk around the display, but do not require it. Out-of-viewpoint operations allow the users to interact from different areas by reaching, without changing their viewpoints. Furthermore, in Chapters 7 and 8, we provided functionality to rotate the content of the volumetric display about the vertical axis, to reduce the need to physically walk around the display. This could be even more important functionality if the user had a restricted viewing angle, and could not obtain a desired viewpoint by physically moving. However, our usage observations in Chapter 8 indicate that users often prefer to physically walk around the display, further validating the benefits of providing a 360° viewing angle.

9.3.4 Enclosed Display Volume

This property is probably the most critical to the interaction techniques which we have designed, since it is a main property that distinguishes volumetric displays from virtual reality [Balakrishnan et al. 2001]. It is possible that future implementations of volumetric displays will allow users to reach directly into the display volume. To date, the only volumetric display which we are aware that does not possess a physical enclosure is a laser plasma display which projects imagery in midair [Uchiyama et al. 2006]. However, due to the extreme heat of the laser plasma, users are still unable to reach in and touch the imagery.

Even without the physical barrier, volumetric displays would still be different from virtual reality environments, due to their limited display volume. As such, the results presented in this thesis would still likely take precedence over results from virtual reality research. However, a number of our results would need to be reconsidered. For example, with the ability to reach in and grab objects, the best method of selection *may* become direct touch. However, this may not necessarily be true, since direct touch would be physically demanding for the user, and it could be the case that the depth ray would still perform best and be preferred by users. However, a study to investigate this would certainly be necessary. Moreover, the interactions developed in Chapter 7, where the user gestured on the surface of the display would no longer be possible. This could actually be a drawback, if users no longer had a tactile surface to interact with [Hinckley et al. 1994]. It may become necessary to introduce some sort of auxiliary interaction surface, such as a flat panel display, for 2D user interface operations, such as making selections from menus. Interface operations which are 3D in nature, such as object manipulations, could be carried out by directly grabbing and manipulating the objects.

9.4 Considerations for Other Display Forms

In this thesis, we assumed volumetric displays to posses the above four general display properties. However, our experiments and implementations were all carried out on a single volumetric display, with a specific display form. It is thus necessary to consider the applicability of our results to other display forms, even if they satisfy the four main properties. In the previous section, we considered how our results would change if the four indentified properties did not hold. Here we consider the impact on our results if the properties were still maintained, but the form of the display varied. Possible variations in display form include size, shape, and occlusion capabilities.

9.4.1 Display Size

The size of volumetric displays could have an important impact on how the display will be used. Our display fit on top of a small table. This display size provides a similar viewing angle as a small monitor, and is large enough that numerous users can easily stand around the display to view and interact with imagery. As long as a volumetric display has a comparable size to this, we expect that all of our results can be directly applied.

Displays as small as sugar cubes have been developed [Downing et al. 1996]. If the display being used was that small, it would likely become a personal display device, rather than a collaborative one. One interesting scenario would be using such a small display as a "virtual snow globe", and passing it around in a meeting while discussing a 3D model. At such a small size, it becomes more likely that the displayed imagery would be static or non-interactive. However, simple gestures, such as scrubbing the surface or tapping on the surface could be used to perform simple operations, such as to view different models, or to rotate or scale the model. The device could also have small physical buttons on its base, such as those found on a digital camera, for interacting with the imagery, or setting display modes.

While we are unaware of any current examples today, it is possible that significantly larger volumetric displays could also be developed in the future. It would be interesting to consider a display large enough to display a one-to-one representation of a car, for example. Such a display would require new human factors studies, and a reconsideration of some of the techniques which we developed.

For example, with such a large display, it would be interesting to complement our depth perception study presented in Chapter 3, with a study evaluating sense of presence. A large display could provide a high viewing angle, similar to a wide-screen monitor. This could increase a user's feeling of being inside a scene, rather than looking at it from an outside point of view. In Chapter 8, we received feedback from an architect that an increased viewing angle and sense of presence would be particularly beneficial.

It would also be useful to reexamine a user's ability to point at 3D targets, which we studied in Chapter 5, when targets were much further away. As the distance between a 3D target and the user increased, the perpetual requirements to point inside of that target would likely increase as well. Understanding the nature of this effect would be important for understanding how selection techniques should be developed.

The techniques which we developed would also require reconsideration if the display were significantly larger. Performing selections, for example, would involve the challenge of reaching distant objects. If the depth ray were to be used, a new mapping would be required for the depth marker, so that it could reach both the front and the back of the display. New mappings would also be required for 3D cursors, such as those used in Chapter 8. The mappings would likely need to be relative, with clutches being required to reach all areas of the volume.

New techniques could also be considered. With the current size of volumetric displays, even with the enclosure, the user feels like all imagery is within reach. However with a larger display, this would no longer be true. A user may not feel comfortable interacting with an aspect of the imagery that is on the opposite end, or even in the center, of the display. Techniques from bringing remote objects closer to users could be explored.

A benefit of a larger display would be that each user would have more display space for their own personalized spaces and interface widgets. Heads-up textual labels would be more appropriate, as they would cause less clutter between users. Menus and dialog boxes could be displayed directly in front of the appropriate user, with little chance of interfering with other uses. Personalized volumes or windows, such as those explored in Chapter 8 would also have more value.

9.4.2 Display Shape

It is interesting to note the various shapes of volumetric displays which have been proposed or developed. These include cuboid, cylindrical, hemispheric, and even conic. However, the shape of volumetric displays has little relevance to our work. We do not foresee any effect of the display shape on any of our lower-level studies on perceptual and motor capabilities. The only real effect may be on the techniques which require direct interaction with the surface of the display, which we explored in Chapter 7. However, when we consider these techniques, none had any reliance on the shape of the enclosure. One minor issue is that for techniques which require scrubbing the surface, it may be awkward if the user had to pass over a sharp corner, such as the edge of a cube. Such scrubbing was required for rotating an object and rotating the surface browser. While the system should allow passing over corners, if the associated interaction exists on the surface, the techniques should also be designed to allow for clutching, so that the user is never forced to pass over such edges.

One other interesting effect which could be studied is if a cube-shaped enclosure would cause users to form spatial territories when collaborating. With a hemispheric shape, there is no clear front, back, or sides of the display. In our usage observations in Chapter 8, users walked freely around the display. With a cube-shaped enclosure, users may feel that they should be positioned directly in front of one of the faces of the cube.

9.4.3 Occlusion Capabilities

The light which current volumetric displays emit is omnidirectional, as it can be seen from any viewpoint around the displays. As a result, viewpoint dependent rendering effects are not possible. Most notably, displays are incapable of performing any sort of object occlusion. Efforts in developing occlusion capable displays have begun [Cossairt et al. 2007, Jones et al. 2007], including our own work presented in Chapter 8. Aside from increasing the overall viewing experience, the ability to occlude objects would have little effect on the interaction issues which we explored for volumetric displays. It would be interesting to evaluate the extent to which occlusions increased the viewing quality, by performing perceptual studies similar to our experiment in Chapter 3. The only other effect occlusion capabilities may have is on our selection techniques. We assumed that users could always see their desired goal target, since all objects are always visible when occlusion isn't possible. Outside of this thesis work, we have started to explore techniques for managing object occlusions during selection [Vanacken et al. 2007], which would become applicable.

9.5 Limitations

In this section we outline some of the limitations associated with the work presented in this thesis.

One issue across all of our studies is that participants generally came from a pool of undergraduate computer science students. As such, participants probably had higher than average technical skills and experiences. It would thus be difficult to generalize our results to less technical users. However, this pool of participants that we did use is probably representative to the types of users who would be working with volumetric displays in the future.

The number of participants which were used should also be discussed. All of our studies used 10-12 participants. While this is fairly typical for studies in the Human-Computer Interaction literature, it is a relatively low number of participants. The decision to use a lower number of participants was mostly due to available resources. However, we feel that this is not a limitation of our work, since our analysis of these studies did produce statistically significant results.

Another limitation related to our study design was the use of Latin Squares when there were more participants than conditions. Our strategy was to use a balanced Latin Square, and to repeat each row of the Latin Square an equal number of times across multiple participants. This design ensured our experiment remained balanced, but an alternative approach would be to use multiple Latin Squares so more orderings were used across the entire participant pool. We refer the reader to Box et al. [1978] for a discussion of other possible strategies which could be used.

An issue related to the results of our studies is that some error rates were very high. For example, in the experiment reported in Section 5.4, error rates were widely different for low and high values of the target depth. It thus makes it somewhat difficult to formally analyze movement times in these different conditions, and to have a model which uniformly addresses all conditions. An alternative approach would be to separately analyze conditions which have significantly different error rates. Similarly, in Chapter 3, a number of the tasks had high associated error rates. If the tasks were easier, we may have obtained a different pattern of results.

A related issue is that the volumetric display technology which we used has a number of quality artifacts, resulting in reduced performance in specific conditions (for example, see Figure 3-9). This probably had an impact on all of our studies, but in particular, was problematic in our study in Chapter 3 comparing display technologies. While the results of the study were informative and important, the same type of study will need to be carried out again when the problematic artifacts are addressed.

A more specific issue, relevant only to the study in Chapter 3, is that we did not investigate if the differing display technologies had different learning rates. This was omitted from our analysis because our experiment design did not include multiple blocks of trials. This would be an important effect to study in the future.

9.6 Future Directions

In this thesis we have tried to isolate and investigate the core human factor questions and interaction issues, in an attempt to the lay the groundwork for future research and application development with volumetric displays. Our work provides some important initial steps towards making volumetric displays interactive 3D platforms. Through our research we have identified a number of future areas of research which we now describe.

9.6.1 Working Applications

While we developed a number of working user interfaces, our developments were restricted to prototypes which served more as platforms for explorations into user interfaces and interaction techniques. We chose to do this, rather than building working applications which could be used in practice for two reasons. First, we felt that before building such real-applications, it would be important to first grasp an understanding of the related interaction issues, and to develop a set of appropriate interaction techniques based on a sound understanding of the user's capabilities. Second, the current display quality of volumetric displays makes it impossible to develop a working application which could actually be used in practice for any sort of real-world task.

However, now that we have investigated the important interaction issues, and as display quality improves, it will be possible to develop working applications for volumetric displays. Once this can be done, it will be interesting to deploy such applications in real environments, to assess their utility. The possible application domains are almost endless, however some particular applications of interest include:

- *Medical Diagnosis*: Volumetric displays could be used to view imagery obtained through x-rays, MRIs and CAT scans. Doctors could interactively view the imagery within its originating 3D structure, rather than viewing the imagery slice by slice as a series of 2D images. Functionality could be provided to cull out occluding structures, and to cut out and manipulate surfaces to highlight the areas of interest.
- *Surgical Planning*: This application is one of the recommendations which came

from our interviews with anatomy experts presented in Chapter 8. In surgical planning, surgeons must consider the relative positioning of complex 3D structures in planning the best approach to treating or accessing a specific internal structure. Carrying out this planning on a volumetric display would allow the surgeons to view these structures in true 3D space. The volumetric display could be used to interactively highlight and compare various routes to the relevant areas, and indicate possible problematic approaches.

- *Educational Tool*: The volumetric display could be used as a 3D model viewer for educational purposes in classroom settings. For example, a high school chemistry class could have a volumetric display in the back of the classroom, and students could use the display to view new structures which they were studying. An application similar to the one developed in Chapter 8 could be used. The display could be used to store thousands of models, allowing students to browse models which possess similarities, and the display could label different areas of interest, depending on what the students were currently studying.
- *Design Studio Display*: The volumetric displays could also be used as a somewhat public terminal in social spaces, supporting quick and informal interactions. For example, the display could be placed in the lobby of a design studio, so designers walking by could quickly view and browse the models at the studio which were currently being developed.

The above is just a sample of possible applications for volumetric displays. In all of the above cases, it would not only be interesting to develop the associated applications, but to also study them in use. For applications such as surgical planning, abstracted experimental tasks could be used to quantitatively compare the volumetric displays to other current practices for accomplishing the tasks. For the other applications, such as the design studio display, more qualitative, sociology type studies could be performed. The application could be deployed in actual design studios, and the usage practices which developed could be studied, to see how the display was leveraged.

9.6.2 Input and Ergonomics

In developing interactions for volumetric displays, we explored several input configurations, such as hand and finger gestures, and six degree-of-freedom input devices. However, our main focus was on the interaction techniques which these input devices enabled, and not the input devices themselves. A future line of work is to study input devices for volumetric displays, to understand what is the most appropriate for the various usage scenarios, such as those outlined above. One type of input which we did not explore is a standard mouse, or a mouse enhanced with extra degrees of freedom. Such a device would have the advantage of being operated on a surface, not requiring the user to work in midair, which could be tiring.

Related to this issue, we have not studied the ergonomics associated with interactive volumetric display applications. If such displays are going to be used for real-world tasks, it is important to know if there will be any ill-effects caused by working with the display for a long period of time. Fatigue from operating a device in midair is just one example. Other possible issues could arise from any noticeable screen flicker, or from constant auditory humming caused by the display. While our observations tend to indicate that this will not be a problem, it would be important to investigate these issues further before deploying volumetric displays for widespread use.

9.7 Conclusion

We have presented a thorough exploration of the issues associated in transitioning volumetric displays from static visualization devices to interactive platforms for 3D applications. We first presented a series of low-level human factor studies, allowing us to subsequently build new interaction techniques and user interfaces based on underlying theoretical models and experimental data. As this thesis is the first in-depth investigation of its type, it was not possible to exhaust all possible topics, and in the above section we have outlined relevant lines of future work, that can be pursued by ourselves and others. However, we feel that were able to focus on core human factors and interaction issues, allowing us to lay the groundwork for the development of interactive volumetric display applications. Our work can thus serve as a baseline for future experiments, theoretical models, interaction techniques, and user interfaces to build upon.

References

"Things could be worse. Suppose your errors were counted and published every day, like those of a baseball player."

\- Unknown

[Accot and Zhai 1997] Accot, J. and S. Zhai (1997). Beyond Fitts' Law: Models for trajectory-based HCI tasks. *ACM CHI Conference on Human Factors in Computing Systems*, 295-302.

[Accot and Zhai 2003] Accot, J. and S. Zhai (2003). Refining Fitts' law models for bivariate pointing. *ACM CHI Conference on Human Factors in Computing Systems*, 193-200.

[Agrawala et al. 1997] Agrawala, M., A. C. Beers, I. McDowall, B. Frohlich, M. Bolas and P. Hanrahan (1997). The two-user Responsive Workbench: support for collaboration through individual views of a shared space. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, 327-332.

[Arditi 1986] Arditi, A. (1986). Binocular vision. *Handbook of Perception and Human Performance*, New York: John Wiley and Sons.

[Arthur et al. 1993] Arthur, K., C. Ware and K. Booth (1993). Evaluating 3D task performance for fish-tank virtual worlds. *ACM Transactions on Information Systems*, 11(3): 239-265.

[Baecker 1994] Baecker, R. M. (1994). Readings in GroupWare and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration, Morgan Kaufmann Publishers Inc.

[Baecker et al. 1994] Baecker, R. M., G. Glass, A. Mitchell and I. Posner (1994). SASSE: the collaborative editor. *ACM CHI Conference on Human Factors in Computing Systems, video companion*.

[Bahr et al. 1996] Bahr, D., K. Langhans, M. Gerken, C. Vogt, D. Bezecny and D. Homann (1996). FELIX: A volumetric 3D laser display. *SPIE Projection Displays*, 265-273.

[Balakrishnan et al. 1997] Balakrishnan, R., T. Baudel, G. Kurtenbach and G. Fitzmaurice (1997). The Rockin'Mouse: Integral 3D manipulation on a plane. *ACM CHI Conference on Human Factors in Computing Systems*, 311-318.

[Balakrishnan et al. 1999] Balakrishnan, R., G. Fitzmaurice, G. Kurtenbach and K. Singh (1999). Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. *ACM I3D Symposium on Interactive 3D Graphics*, 111-118.

[Balakrishnan and Kurtenbach 1999] Balakrishnan, R. and G. Kurtenbach (1999). Exploring bimanual camera control and object manipulation in 3D graphics interfaces. *ACM CHI Conference on Human Factors in Computing Systems*, 56-63.

[Balakrishnan et al. 2001] Balakrishnan, R., G. Fitzmaurice and G. Kurtenbach (2001). User interfaces for volumetric displays. *IEEE Computer*, 34(3): 37-45.

[Barfield et al. 1997] Barfield, W., C. Hendrix and K. Bystrom (1997). Visualizing the structure of virtual objects using head tracked stereoscopic displays. *Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)*,

[Baudel and Beaudouin-Lafon 1993] Baudel, T. and M. Beaudouin-Lafon (1993). Charade: remote control of objects using free-hand gestures. *Communications of the ACM*, 36(7): 28-35.

[Bell et al. 2001] Bell, B., S. Feiner and T. Hollerer (2001). View management for virtual and augmented reality. *ACM UIST Symposium on User Interface Software and Technology*, 101-110.

[Benko et al. 2006] Benko, H., A. Wilson and P. Baudisch (2006). Precise Selection Techniques for Multi-Touch Screens. *ACM CHI Conference on Human Factors in Computing Systems*, 1263-1272.

[Bier and Freeman 1991] Bier, E. A. and S. Freeman (1991). MMM: a user interface architecture for shared editors on a single screen. *ACM UIST Symposium on User Interface Software and Technology*, 79-86.

[Blanch et al. 2004] Blanch, R., Y. Guiard and M. Beaudouin-Lafon (2004). Semantic pointing: improving target acquisition with control-display ratio adaptation. *ACM CHI Conference on Human Factors in Computing Systems*, 519-525.

[Blundell et al. 1993] Blundell, B. G., A. J. Schwarz and D. K. Horrell (1993). Volumetric three-dimensional display systems - their past, present and future. *IEEE Science and Engineering Education Journal*, 2(5): 196-200.

[Boritz et al. 1991] Boritz, J., K. Booth and W. Cowan (1991). Fitts' law studies of directional mouse movement. *Graphics Interface*, 216-223.

[Bowman and Hodges 1997] Bowman, D. and L. Hodges (1997). An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. *ACM I3D Symposium on Interactive 3D Graphics*, 35-38.

[Bowman and Hodges 1999] Bowman, D. and L. Hodges (1999). Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. *The Journal of Visual Languages and Computing*, 10(1): 37-53.

[Bowman et al. 2003] Bowman, D., C. North, J. Chen, N. Polys, P. Pyla and U. Yilmaz (2003). Information-rich virtual environments: theory, tools, and research agenda. *ACM VRST Symposium on Virtual Reality Software and Technology*, 81-90.

[Bowman et al. 1999] Bowman, D. A., D. B. Johnson and L. F. Hodges (1999). Testbed evaluation of virtual environment interaction. *ACM VRST Symposium on Virtual Reality Software and Technology*, 26-33.

[Bowman and Wingrave 2001] Bowman, D. A. and C. A. Wingrave (2001). Design and Evaluation of Menu Systems for Immersive Virtual Environments. *Proceedings of the Virtual Reality 2001 Conference (VR'01)*, 149.

[Box et al. 1978] Box, G. E. P., W. G. Hunter, J. S. Hunter and W. G. Hunter (1978). Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building, John Wiley & Sons.

[Brooks 1988] Brooks, F. (1988). Grasping reality through illusion - Interactive graphics serving science. *ACM CHI Conference on Human Factors in Computing Systems*, 1-11.

[Budinger 1984] Budinger, T. F. (1984). An Analysis of 3D Display Strategies. *SPIE Processing and Display of Three Dimensional Data*, 2-8.

[Burtnyk et al. 2002] Burtnyk, N., A. Khan, G. Fitzmaurice, R. Balakrishnan and G. Kurtenbach (2002). StyleCam: Interactive stylized 3D navigation using integrated spatial and temporal controls. *ACM UIST Symposium on User Interface Software and Technology*, 101-110.

[Butterworth et al. 1992] Butterworth, J., A. Davidson, S. Hench and M. T. Olano (1992). 3DM: a three dimensional modeler using a head-mounted display. *ACM I3D Symposium on Interactive 3D Graphics*, 135-138.

[Buxton and Fitzmaurice 1998] Buxton, W. and G. W. Fitzmaurice (1998). HMD's, Caves, & Chameleon: A human-centric analysis of interaction in virtual space. *Computer Graphics, The SIGGRAPH Quarterly*, 32(4): 64-68.

[Card et al. 1991] Card, S. K., J. D. MacKinlay and G. G. Robertson (1991). A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems*, 9: 99-122.

[Chen et al. 2003] Chen, J., P. S. Pyla and D. A. Bowman (2003). Testbed Evaluation of Navigation and Text Display Techniques in an Information-Rich Virtual Environment, Computer Science, Virginia Tech.

[Clark 1976] Clark, J. H. (1976). Designing surfaces in 3-D. *Communications of the ACM*, 19(8): 454-460.

[Conner et al. 1992] Conner, B., S. S. Snibbe, K. P. Herndon, D. Robbins, R. Zeleznik and A. v. Dam (1992). Three dimensional widgets. *Computer Graphics*, 22(4): 121-129.

[Cossairt et al. 2007] Cossairt, O. S., J. Napoli, S. L. Hill, R. K. Dorval and G. E. Favalora (2007). Occlusion-capable multiview volumetric three-dimensional display. *Applied Optics*, 46(8): 1244-1250.

[Crossman 1956] Crossman, E. (1956). The measurement of perceptual load in manual operations, University of Birmingham.

[Cruz-Neira et al. 1992] Cruz-Neira, C., D. J. Sandin, T. A. DeFanti, R. V. Kenyon and J. C. Hart (1992). The CAVE: Audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6): 65-72.

[Cutler et al. 1997] Cutler, L., B. Frohlich and P. Hanrahan (1997). Two-handed direct manipulation on the responsive workbench. *ACM I3D Symposium on Interactive 3D Graphics*, 107-113.

[Czernuszenko et al. 1997] Czernuszenko, M., D. Pape, D. Sandin, T. DeFanti, G. L. Dawe and M. D. Brown (1997). The ImmersaDesk and infinity wall projection based virtual reality displays. *Computer Graphics*, 31(2): 46-49.

[DeLucia 1995] DeLucia, P. R. (1995). Effects of pictorial relative size and ground-intercept information on judgments about potential collisions in perspective displays. *Human Factors*, 37: 528-538.

[Diamond et al. 1982a] Diamond, R., A. Wynn, K. Thomsen and J. Turner (1982a). Three- Dimensional Perception for One-Eyed Guys. *Computational Crystallography*, Oxford, Clarendon Press.

[Diamond et al. 1982b] Diamond, R., A. Wynn, K. Thomsen and J. Turner (1982b). Three-dimensional perception for one-eyed guys. *Computational Crystallography*, Oxford, Clarendon Press.

[Dietz and Leigh 2001] Dietz, P. and D. Leigh (2001). DiamondTouch: a multi-user touch technology. *ACM UIST Symposium on User Interface Software and Technology*, 219-226.

[Djajadiningrat et al. 1997] Djajadiningrat, J. P., G. J. F. Smets and C. J. Overbeeke (1997). Cubby: A multiscreen movement parallax display for direct manual manipulation. *Displays*, 17: 191-197.

[Dodgson 2005] Dodgson, N. A. (2005). Autostereoscopic 3D Displays. *Computer*, 38(8): 31-36.

[Downing et al. 1996] Downing, E., L. Hesselink, J. Ralston and R. Macfarlane (1996). A Three-Color, Solid-State, Three-Dimensional Display. *Science*, 273(5279): 1185-1189.

[Ebert et al. 1999] Ebert, D., E. Bedwell, S. Maher, L. Smoliar and E. Downing (1999). Realizing 3D visualization using crossed-beam volumetric displays. *Communications of the ACM*, 42(8): 101-107.

[Ellis et al. 1991] Ellis, C. A., S. J. Gibbs and G. Rein (1991). Groupware: some issues and experiences. *Communications of the ACM*, 34(1): 39-58.

[Favalora et al. 2002] Favalora, G. E., J. Napoli, D. M. Hall, R. K. Dorval, M. Giovinco, M. J. Richmond and W. S. Chun (2002). 100-million-voxel volumetric display. *SPIE - Volume 4712*, 300-312.

[Favalora 2005] Favalora, G. E. (2005). Volumetric 3D Displays and Application Infrastructure. *Computer*, 38(8): 37-44.

[Feiner et al. 1993a] Feiner, S., B. MacIntyre, M. Haupt and E. Solomon (1993a). Windows on the world: 2D windows for 3D augmented reality. *ACM UIST Symposium on User Interface Software and Technology*, 145-155.

[Feiner et al. 1993b] Feiner, S., B. Macintyre and D. Seligmann (1993b). Knowledge-based augmented reality. *Communications of the ACM*, 36(7): 53-62.

[Fisher 1982] Fisher, S. (1982). Viewpoint dependent imaging: An interactive stereoscopic display. *Processing and Display of Three-Dimensional Data II*, 41-45.

[Fitts 1954] Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6): 381-391.

[Fitzmaurice et al. 1999] Fitzmaurice, G., R. Balakrishnan and G. Kurtenbach (1999). An exploration into supporting artwork orientation in the user interface. *ACM CHI Conference on Human Factors in Computing Systems*, 167-174.

[Fitzmaurice 1993] Fitzmaurice, G. W. (1993). Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7): 38-49.

[Foley et al. 1990] Foley, J. D., A. v. Dam, S. K. Feiner and J. F. Hughes (1990). Computer Graphics: Principles and Practice, Addison-Wesley Professional.

[Forsberg et al. 1996] Forsberg, A., K. Herndon and R. Zeleznik (1996). Aperture based selection for immersive virtual environments. *ACM UIST Symposium on User Interface Software and Technology*, 95-96.

[Frohlich and Plate 2000] Frohlich, B. and J. Plate (2000). The cubic mouse: a new device for three-dimensional input. *ACM CHI Conference on Human Factors in Computing Systems*, 526-531.

[Goble et al. 1995] Goble, J. C., K. Hinckley, R. Pausch, J. W. Snell and N. F. Kassell (1995). Two-Handed Spatial Interface Tools for Neurosurgical Planning. *Computer*, 28(7): 20-26.

[Greenberg and Marwood 1994] Greenberg, S. and D. Marwood (1994). Real time groupware as a distributed system: concurrency control and its effect on the interface. *ACM CSCW Conference on Computer Supported Cooperative Work*, 207-217.

[Grosjean et al. 2002] Grosjean, J., J. M. Burkhardt, S. Coquillart and P. Richard (2002). Evaluation of the Command and Control Cube. *IEEE International Conference on Multimodal Interfaces*, 473-478.

[Grossman et al. 2003] Grossman, T., R. Balakrishnan and K. Singh (2003). An interface for creating and manipulating curves using a high degree-of-freedom input device. *ACM CHI Conference on Human Factors in Computing Systems*, 185-192.

[Grossman and Balakrishnan 2004] Grossman, T. and R. Balakrishnan (2004). Pointing at trivariate targets in 3D environments. *ACM CHI Conference on Human Factors in Computing Systems*, 447-454.

[Grossman et al. 2004] Grossman, T., D. Wigdor and R. Balakrishnan (2004). Multi finger gestural interaction with 3D volumetric displays. *ACM UIST Symposium on User Interface Software and Technology*, 61-70.

[Grossman and Balakrishnan 2005a] Grossman, T. and R. Balakrishnan (2005a). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *ACM CHI Conference on Human Factors in Computing Systems*, 281-290.

[Grossman and Balakrishnan 2005b] Grossman, T. and R. Balakrishnan (2005b). A probabilistic approach to modeling 2D pointing. *ACM Transactions on Computer-Human Interaction*, 12(3): 435-459.

[Grossman and Balakrishnan 2006a] Grossman, T. and R. Balakrishnan (2006a). An evaluation of depth perception on volumetric displays. *Proceedings of the Working Conference on Advanced Visual Interfaces*, 193-200.

[Grossman and Balakrishnan 2006b] Grossman, T. and R. Balakrishnan (2006b). The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. *ACM UIST Symposium on User Interface Software and Technology*, 3-12.

[Grossman et al. 2007] Grossman, T., D. Wigdor and R. Balakrishnan (2007). Exploring and Reducing the Effects of Orientation on Text Readability in Volumetric Displays. *ACM CHI Conference on Human Factors in Computing Systems*, 483-492.

[Grossman and Balakrishnan 2008] Grossman, T. and R. Balakrishnan (2008). Collaborative interaction with volumetric displays. *ACM CHI Conference on Human Factors in Computing Systems*, 383-392.

[Guiard 1987] Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19(4): 486-517.

[Guiard et al. 2004] Guiard, Y., R. Blanch and M. Beaudouin-Lafon (2004). Object pointing: a complement to bitmap pointing in GUIs. *Graphics Interface*, 9-16.

[Hinckley et al. 1994] Hinckley, K., R. Pausch, J. C. Goble and N. Kassell (1994). A survey of design issues in spatial input. *ACM UIST Symposium on User Interface Software and Technology*, 213-222.

[Hinckley et al. 1997] Hinckley, K., J. Tulio, R. Pausch, D. Proffitt and N. Kassell (1997). Usability analysis of 3D rotation techniques. *ACM UIST Symposium on User Interface Software and Technology*, 1-10.

[Hinckley et al. 1998] Hinckley, K., R. Pausch, D. Proffitt and N. Kassell (1998). Two-handed virtual manipulation. *ACM Transactions on Computer-Human Interaction*, 5(3): 260-302.

[Hoffmann and Sheikh 1994] Hoffmann, E. and I. Sheikh (1994). Effect of varying target height in a Fitts movement task. *Ergonomics*, 37(6): 1071-1088.

[Hofstadter 1979] Hofstadter, D. R. (1979). Goedel, Escher, Bach: an Eternal Golden Braid, NY: Basic Books.

[Hollerbach and Jacobsen 1993] Hollerbach, J. M. and S. C. Jacobsen (1993). Optimized fingertip mapping: a general algorithm for robotic hand teleoperation. *Presence*, 1.2(3): 203-220.

[Hubona et al. 1999] Hubona, G., P. Wheeler, G. Shirah and M. Brandt (1999). The relative contributions of stereo, lighting, and background scenes in promoting 3D depth visualization. *ACM Transactions on Computer-Human Interaction*, 6(3): 214-242.

[ITT-Laboratories 1960] ITT-Laboratories (1960). New Display Gives Realistic 3D Effect. *Aviation Week*: 66-67.

[Jones et al. 2007] Jones, A., I. McDowall, H. Yamada, M. Bolas and P. Debevec (2007). Rendering for an interactive 360° light field display. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, Article No. 40.

[Kabbash and Buxton 1995] Kabbash, P. and W. Buxton (1995). The "Prince" technique: Fitts' law and selection using area cursors. *ACM CHI Conference on Human Factors in Computing Systems*, 273-279.

[Ketchpel 1963] Ketchpel, R. D. (1963). Direct-View Three-Dimensional Display Tube. *IEEE Transactions on Electron Devices*: 324-328.

[Kim et al. 1996] Kim, I. I., E. J. Korevaar and H. Hakakha (1996). Three-dimensional volumetric display in rubidium vapor. *SPIE Projection Displays*, 274-284.

[Kim 1981] Kim, S. (1981). Inversions, Byte Books.

[Krueger 1991] Krueger, M. (1991). VIDEOPLACE and the interface of the future. *The art of human computer interface design*. B. Laurel. Menlo Park, CA, Addison Wesley**:** 417-422.

[Kruger et al. 2003] Kruger, R., S. Carpendale, S. D. Scott and S. Greenberg (2003). How people use orientation on tables: comprehension, coordination and communication. *ACM SIGGROUP Conference on Supporting Group Work*, 369-378.

[Kurtenbach and Buxton 1993] Kurtenbach, G. and W. Buxton (1993). The limits of expert performance using hierarchical marking menus. *ACM CHI Conference on Human Factors in Computing Systems*, 35-42.

[Kurtenbach et al. 1997] Kurtenbach, G., G. Fitzmaurice, T. Baudel and W. Buxton (1997). The design of a GUI paradigm based on tablets, two-hands, and transparency. *ACM CHI Conference on Human Factors in Computing Systems*, 35-42.

[Lang et al. 2003] Lang, J., T. Ligon and D. Hilligoss (2003). The OmniGlobe: A Self-Contained Spherical Display System. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques (Emerging Technologies)*.

[Langhans et al. 1998] Langhans, K., D. Bezecny, D. Homann, D. Bahr, C. Vogt, C. Blohm and K.-H. Scharschmidt (1998). New portable FELIX 3D display. *SPIE Projection Displays*, 204-216.

[Langolf et al. 1976] Langolf, G. D., D. B. Chaffin and J. A. Foulke (1976). An investigation of Fitts' law using a wide range of movement amplitudes. *Journal of Motor Behaviour*, 8: 113-128.

[Larson et al. 2000] Larson, K., M. v. Dantzich, M. Czerwinski and G. Robertson (2000). Text in 3D: some legibility results. *ACM CHI Conference on Human Factors in Computing Systems*, 145-146.

[Lewis et al. 1971] Lewis, J. D., C. M. Verber and R. B. McGhee (1971). A True Three-Dimensional Display. *IEEE Transactions on Electron Devices*, 18: 724-732.

[Liang and Green 1993] Liang, J. and M. Green (1993). Quick Prototyping for Engineering Design. *Computer Applications and Design Abstraction*, 157- 162.

[Liang and Green 1994] Liang, J. and M. Green (1994). JDCAD: A highly interactive 3D modeling system. *Computers and Graphics*, 18(4): 499-506.

[Lipscomb 1981] Lipscomb, J. S. (1981). Three-dimensional cues for a molecular computer graphics system. 238.

[Lipton 1985] Lipton, L. (1985). Stereoscopic Television System. United States.

[MacKenzie 1992] MacKenzie, S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7: 91-139.

[MacKenzie and Buxton 1992] MacKenzie, S. and W. Buxton (1992). Extending Fitts' law to two-dimensional tasks. *ACM CHI Conference on Human Factors in Computing Systems*, 219-226.

[Mackinlay et al. 1990] Mackinlay, J., S. Card and G. Robertson (1990). Rapid controlled movement through a virtual 3D workspace. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, 171-176.

[McAllister 1993] McAllister, D. F. (1993). Stereo Computer Graphics and other True 3D Technologies, Princeton, New Jersey: Princeton University Press.

[McCauley and Sharkey 1992] McCauley, M. E. and T. J. Sharkey (1992). Cybersickness: Perception of self-motion in virtual environments. *Presence*, 1(3): 311-318.

[McGuffin and Balakrishnan 2002] McGuffin, M. and R. Balakrishnan (2002). Acquisition of expanding targets. *ACM CHI Conference on Human Factors in Computing Systems*, 57-64.

[McGuffin et al. 2003] McGuffin, M., L. Tancau and R. Balakrishnan (2003). Using deformations for browsing volumetric data. *IEEE Visualization*, 401-408.

[McKenna 1992a] McKenna, M. (1992a). Interactive viewpoint control and three-dimensional operations. *ACM Symposium on Interactive 3D Graphics*, 53-56.

[McKenna 1992b] McKenna, M. (1992b). Interactive viewpoint control and three-dimensional operations. *ACM I3D Symposium on Interactive 3D Graphics*, 53-56.

[Mine 1995a] Mine, M. (1995a). Virtual environment interaction techniques, UNC Chapel Hill Computer Science Technical Report TR95-018.

[Mine 1995b] Mine, M. (1995b). ISAAC: A virtual environment tool for the interactive construction of virtual worlds, UNC Chapel Hill Computer Science Technical Report TR95-020.

[Mine et al. 1997] Mine, M., F. Brooks and C. Sequin (1997). Moving objects in space: exploiting proprioception in virtual environment interaction. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, 19-26.

[Montgomery 2001] Montgomery, D. C. (2001). Design and Analysis of Experiments. New York, NY, John Wiley and Sons.

[Morris et al. 2004] Morris, M. R., K. Ryall, C. Shen, C. Forlines and F. Vernier (2004). Beyond "social protocols": multi-user coordination policies for co-located groupware. *ACM CSCW Conference on Computer Supported Cooperative Work*, 262-265.

[Murata 1999] Murata, A. (1999). Extending effective target width in Fitts' law to a two-dimensional pointing task. *International Journal of Human-Computer Interaction*, 11(2): 137-152.

[Narayan et al. 2005] Narayan, M., W. Leo, Z. Xiaoyu, B. Pradyut and B. Doug (2005). Quantifying the benefits of immersion for collaboration in virtual environments. Proceedings of the ACM symposium on Virtual reality software and technology. Monterey, CA, USA, ACM.

[Neale 1998] Neale, D. (1998). Head-Mounted Displays: Product Reviews and Related Design Considerations, Human-Computer Interaction Laboratory, Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061-0118.

[Nunamaker et al. 1991] Nunamaker, J. F., A. R. Dennis, J. S. Valacich, D. Vogel and J. F. George (1991). Electronic meeting systems. *Communications of the ACM*, 34(7): 40-61.

[Olwal and Feiner 2003] Olwal, A. and S. Feiner (2003). The Flexible Pointer - An Interaction Technique for Selection in Augmented and Virtual Reality. *ACM UIST Symposium on User Interface Software and Technology (Conference Supplement)*, 81-82.

[Pfautz 2000] Pfautz, J. (2000). Depth Perception in Computer Graphics University of Cambridge.

[Pierce et al. 1997] Pierce, J., A. Forsberg, M. Conway, S. Hong and R. Zeleznik (1997). Image plane interaction techniques in 3D immersive environments. *ACM I3D Symposium on Interactive 3D Graphics*, 39-43.

[Pierce et al. 1999a] Pierce, J., M. Conway, M. v. Dantzich and G. Robertson (1999a). Toolspaces and glances: storing, accessing, and retrieving objects in 3D desktop applications. *ACM CHI Conference on Human Factors in Computing Systems*, 163-168.

[Pierce et al. 1999b] Pierce, J., B. Stearns and R. Pausch (1999b). Two-Handed Manipulation of Voodoo Dolls in Virtual Environments. *ACM I3D Symposium on Interactive 3D Graphics*, 141-145.

[Pierce and Pausch 2002] Pierce, J. S. and R. Pausch (2002). Comparing voodoo dolls and HOMER: exploring the importance of feedback in virtual environments. *ACM CHI Conference on Human Factors in Computing Systems*, 105-112.

[Poston and Serra 1994] Poston, T. and L. Serra (1994). The virtual workbench: dextrous VR. *ACM VRST Virtual Reality Software and Technology*, 111-121.

[Poupyrev et al. 1996] Poupyrev, I., M. Billinghurst, S. Weghorst and T. Ichikawa (1996). The go-go interaction technique: non-linear mapping for direct manipulation in VR. *ACM UIST Symposium on User Interface Software and Technology*, 79-80.

[Ramos et al. 2006] Ramos, G., G. Robertson, M. Czerwinski, D. Tan, P. Baudisch, K. Hinckley and M. Agrawala (2006). Tumble! Splat! helping users access and manipulate occluded content in 2D drawings. *Proceedings of the Working Conference on Advanced Visual Interfaces*, 428-435.

[Rawson 1969] Rawson, E. G. (1969). vibrating varifocal mirrors for 3d imaging. *IEEE Spectrum*: 37-43.

[Rekimoto 1996] Rekimoto, J. (1996). Tilting operations for small screen interfaces. *ACM UIST Symposium on User Interface Software and Technology*, 167-168.

[Rosen et al. 2004] Rosen, P., Z. Pizlo, C. Hoffmann and V. S. Popescu (2004). Perception of 3D spatial relations for 3D displays. *Stereoscopic Displays and Virtual Reality Systems XI*, 9-16.

[Ryall et al. 2004] Ryall, K., C. Forlines, C. Shen and M. R. Morris (2004). Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. *ACM CSCW Conference on Computer Supported Cooperative Work*, 284-293.

[Sachs et al. 1991] Sachs, E., A. Roberts and D. Stoops (1991). 3-draw: A tool for designing 3D shapes. *IEEE Computer Graphics and Applications*, 11(6): 18-26.

[Schafer and Bowman 2004] Schafer, W. and D. Bowman (2004). Evaluating the Effects of Frames of Reference on Spatial Collaboration in Desktop Virtual Environments. *Virtual Reality*, 7(3-4): 164-174.

[Schafer and Bowman 2005] Schafer, W. A. and D. A. Bowman (2005). Integrating 2D and 3D views for spatial collaboration. *ACM SIGGROUP International Conference on Supporting Group Work*, 41-50.

[Schkolne et al. 2001] Schkolne, S., M. Pruett and P. Schroeder (2001). Surface drawing: Creating organic 3D shapes with the hand and tangible tools. *ACM CHI Conference on Human Factors in Computing Systems*, 261-268.

[Scott et al. 2005] Scott, S. D., M. S. T. Carpendale and S. Habelski (2005). Storage Bins: Mobile Storage for Collaborative Tabletop Displays. *IEEE Computer Graphics and Applications*, 25(4): 58-65.

[Sellen et al. 1992] Sellen, A., G. Kurtenbach and W. Buxton (1992). The prevention of mode errors through sensory feedback. *Human Computer Interaction*, 7(2): 141-164.

[Sheikh and Hoffmann 1994] Sheikh, I. and E. Hoffmann (1994). Effect of target shape on movement time in a Fitts task. *Ergonomics*, 37(9): 1533-1548.

[Shen et al. 2004] Shen, C., F. Vernier, C. Forlines and M. Ringel (2004). DiamondSpin: An extensible toolkit for around the table interaction. *ACM CHI Conference on Human Factors in Computing Systems*, 167-174.

[Shepard and Metzler 1971] Shepard, R. and J. Metzler (1971). Mental Rotation of Three-Dimensional Objects. *Science*, 171.

[Shibata et al. 2005] Shibata, T., T. Kawai, M. Otsuki, N. Miyake, Y. Yoshihara and T. Iwasaki (2005). Stereoscopic 3D display with dynamic optical correction for recovering from asthenopia. *Stereoscopic Displays and Virtual Reality Systems XII*, 1-9.

[Shoemaker and Inkpen 2001] Shoemaker, G. and K. Inkpen (2001). Single display privacyware: Augmenting public displays with private information. *ACM CHI Conference on Human Factors in Computing Systems*, 522-529.

[Sollenberger and Milgram 1991] Sollenberger, R. and P. Milgram (1991). A comparative study of rotational and stereoscopic computer graphic depth cues. *Human Factors Society Annual Meeting*, 1452-1456.

[Sollenberger and Milgram 1993] Sollenberger, R. and P. Milgram (1993). Effects of stereoscopic and rotational displays in a three-dimensional path-tracing task. *Human Factors*, 35(3): 483-499.

[Steed and Parker 2004] Steed, A. and C. Parker (2004). 3D Selection Strategies for Head Tracked and Non-Head Tracked Operation of Spatially Immersive Displays. *8th International Immersive Projection Technology Workshop*, May 2004.

[Steed 2006] Steed, A. (2006). Towards a General Model for Selection in Virtual Environments. *IEEE Symposium on 3D User Interfaces*, 103-110.

[Stewart et al. 1999] Stewart, J., B. Bederson and A. Druin (1999). Single display groupware: A model for co-present collaboration. *ACM CHI Conference on Human Factors in Computing Systems*, 286-293.

[Stoakley et al. 1995] Stoakley, R., M. Conway and R. Pausch (1995). Virtual reality on a WIM: Interactive worlds in miniature. *ACM CHI Conference on Human Factors in Computing Systems*, 265-272.

[Sutherland 1968] Sutherland, I. E. (1968). A head-mounted three dimensional display. *Fall Joint Computer Conference*, 757-764.

[Tinker 1965] Tinker, M. (1965). Bases for effective reading, Minneapolis: Lund Press, Inc.

[Tsang et al. 2002] Tsang, M., G. W. Fitzmaurice, G. Kurtenbach, A. Khan and B. Buxton (2002). Boom chameleon: simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. *ACM UIST Symposium on User Interface Software and Technology*, 111-120.

[Tyler et al. 2005] Tyler, T., A. Novobilski, J. Dumas and A. Warren (2005). The Utility of Perspecta 3D Volumetric Display for Completion of Tasks. *Stereoscopic Displays and Virtual Reality Systems XII*, 268-279.

[Uchiyama et al. 2006] Uchiyama, T., K. University and S. Shimada (2006). True 3D Display Using Laser Plasma in the Air. *SIGGRAPH International Conference on Computer Graphics and Interactive Techniques (Emerging Technologies)*.

[van Dam et al. 2000] van Dam, A., A. S. Forsberg, D. H. Laidlaw, J. J. LaViola and R. M. Simpson (2000). Immersive VR for Scientific Visualization: A Progress Report. *IEEE Computer Graphics and Applications*, 20(6): 26-52.

[Van Orden and Broyles 2000] Van Orden, K. F. and J. W. Broyles (2000). Visuospatial Task Performance as a Function of Two- and Three-Dimensional Display Presentation Techniques. *Displays*, 21(1): 17-24.

[Vanacken et al. 2007] Vanacken, L., T. Grossman and K. Coninx (2007). Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. *IEEE Symposium on 3D User Interfaces*, 117-124.

[Venolia 1993] Venolia, D. (1993). Facile 3D direct manipulation. *ACM CHI Conference on Human Factors in Computing Systems*, 31-36.

[Viega et al. 1996] Viega, J., M. J. Conway, G. Williams and R. Pausch (1996). 3D magic lenses. *ACM UIST Symposium on User Interface Software and Technology*, 51-58.

[Vogel and Balakrishnan 2004] Vogel, D. and R. Balakrishnan (2004). Interactive public ambient displays: Transitioning from implicit to explicit, public to personal, interaction with multiple users. *ACM UIST Symposium on User Interface Software and Technology*, 137-146.

[Wanger 1992] Wanger, L. (1992). The effect of shadow quality on the perception of spatial relationships in computer generated imagery. *ACM I3D Symposium on Interactive 3D Graphics*, 39-42.

[Wanger et al. 1992] Wanger, L. C., J. A. Ferwerda and D. P. Greenberg (1992). Perceiving Spatial Relationships in Computer-Generated Images. *IEEE Computer Graphics and Applications*, 12(3): 44-58.

[Ware and Jessome 1988] Ware, C. and D. R. Jessome (1988). Using the bat: a six dimensional mouse for object placement. *IEEE Computer Graphics and Applications*, 8(6): 65-70.

[Ware and Osborne 1990] Ware, C. and S. Osborne (1990). Exploration and virtual camera control in virtual three dimensional environments. *ACM I3D Symposium on Interactive 3D Graphics*, 175-183.

[Ware and Slipp 1991] Ware, C. and L. Slipp (1991). Using velocity control to navigate 3D graphical environments: a comparison of three interfaces. *Human Factors Society 35th Annual Meeting*, 25-32.

[Ware et al. 1993] Ware, C., K. Arthur and K. S. Booth (1993). Fish tank virtual reality. *ACM CHI Conference on Human Factors in Computing Systems*, 37-42.

[Ware and Balakrishnan 1994] Ware, C. and R. Balakrishnan (1994). Reaching for objects in VR displays: Lag and frame rate. *ACM Transactions on Computer-Human Interaction*, 1(4): 331-356.

[Ware and Franck 1996] Ware, C. and G. Franck (1996). Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics*, 15(2): 121-140.

[Ware and Lowther 1997] Ware, C. and K. Lowther (1997). Selection using a one-eyed cursor in a fish-tank VR environment. *ACM Transactions on Computer-Human Interaction*, 4(4): 309-322.

[Wigdor and Balakrishnan 2005] Wigdor, D. and R. Balakrishnan (2005). Empirical Investigation into the Effect of Orientation on Text Readability in Tabletop Displays. *ECSCW European Conference on Computer-Supported Cooperative Work*, 205-224.

[Wigdor et al. 2006] Wigdor, D., C. Shen, C. Forlines and R. Balakrishnan (2006). Effects of display position and control space orientation on user preference and performance. *ACM CHI Conference on Human Factors in Computing Systems*, 309-318.

[Wigdor 2008] Wigdor, D. (2008). The Design of Table-Centric Interactive Spaces *Department of Computer Science,* University of Toronto.

[Worden et al. 1997] Worden, A., N. Walker, K. Bharat and S. Hudson (1997). Making computers easier for older adults to use: area cursors and sticky icons. *ACM CHI Conference on Human Factors in Computing Systems*, 266-271.

[Wu and Balakrishnan 2003] Wu, M. and R. Balakrishnan (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *ACM UIST Symposium on User Interface Software and Technology*, 193-202.

[Wyss et al. 2006] Wyss, H. P., R. Blach and M. Bues (2006). iSith - Intersection-based Spatial Interaction for Two Hands. *IEEE Symposium on 3D User Interfaces*, 59-61.

[Yeh 1993] Yeh, Y.-Y. (1993). Visual and perceptual issues in stereoscopic color displays. *Stereo computer graphics: and other true 3D technologies*, Princeton University Press**:** 50-70.

[Yonas 1979] Yonas, A. (1979). Attached and cast shadows. *Perception and pictorial representation*, C.F. Nodine, Ed. Praeger Publishers**:** 100-109.

[Zanella and Greenberg 2001] Zanella, A. and S. Greenberg (2001). Reducing interference in single-display groupware through transparency. *ECSCW European Conference on Computer-Supported Cooperative Work*, 16-20.

[Zeleznik et al. 1997] Zeleznik, R., A. Forsberg and P. Strauss (1997). Two pointer input for 3D interaction. *ACM I3D Symposium on Interactive 3D Graphics*, 115-120.

[Zeleznik and Forsberg 1999] Zeleznik, R. and A. Forsberg (1999). UniCam - 2D Gestural Camera Controls for 3D Environments. *ACM I3D Symposium on Interactive 3D Graphics*, 169-173.

[Zeleznik and Miller 2005] Zeleznik, R. and T. Miller (2005). Fluid Inking: Modeless Integration of Gestures with Inking. Tech Report CS-05-10, Brown University Computer Science.

[Zhai et al. 1994] Zhai, S., W. Buxton and P. Milgram (1994). The "Silk Cursor": Investigating transparency for 3D target acquisition. *ACM CHI Conference on Human Factors in Computing Systems*, 459-464.

[Zhai 1995] Zhai, S. (1995). Human performance in six degree-of-freedom input control *Department of Industrial Engineering,* University of Toronto.

[Zhai 1998] Zhai, S. (1998). User performance in relation to 3D input device design. *Computer Graphics*, 32(4): 50-54.