A Virtual Environment System for the Comparative Study of Dome and HMD

A Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Jian Chen

May 2002

A Virtual Environment System for the Comparative Study of Dome and HMD

Jian Chen

APPROVED:

Dr. Ernst L. Leiss, Chairman

Dr. R. Bowen Loftin

Dr. Ioannis A. Kakadiaris

Dean, College of Natural
Sciences and Mathematics

ACKNOWLEDGMENTS

First, I am most grateful to Dr. R. Bowen Loftin, for his insight, advice, and guidance over these years and for having given me the opportunity to work on several projects.

I would also like to extend my sincere appreciations to Dr. Ernst L. Leiss who is so generous in his support at various stages, for his thorough review of my thesis, for every valuable comment he has made, and for serving as chairperson of my thesis committee. My thanks also go to Dr. Ioannis A. Kakadiaris for his support in providing office space and machines during this research work.

I am very grateful to Dr. Leiss, Dr. Loftin, and Dr. Kakadiaris for serving on my master thesis committee and for all that they have taught me about doing research.

Special thanks go to Dr. Deborah L. Harm, the project investigator at NASA/JSC, for her continuous support on this project; and to Ms. Taylor Laura, who is in charge of the human subject test on this project, for helping me set up the system at NASA/JSC.

Furthermore, I would like to thank the staff members: in particular, Ching-yao Lin, for his suggestions on many aspects of this research, for always having time to answer my queries and doubts, and for tutoring me in programming skills and in the art of virtual reality technologies; Hector Garcia who used his magic hand to create the beautiful virtual scenario and never tired of providing technical support over these years; and Chau Cao, who set up a completely basic version of the system.

Most of all, I am very grateful to my parents for their love, support, and encouragement. They set a good example for me. And finally, I thank my husband, Meng Ling, for his steadfast support, patience, and love.

A Virtual Environment System for the Comparative Study of Dome and HMD

An Abstract of a Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Jian Chen

May 2002

ABSTRACT

At NASA/JSC, researchers must choose visual display devices that best fit ground training applications in a virtual environment (VE) system. Two major display devices are a 3.7-meter spherical dome equipped with an Elumens™ projector and a VR4® head-mounted display (HMD). They differ in many aspects, such as quality of stereo, field of view, level of immersion, resolution, and the interfaces and interactions. Finally, the related human factor issues, especially subject performance, are not well identified.

A virtual environment system was developed to assist in the comparison of the relative merits of these two different delivery systems based on pick-and-release tasks. The rendering engine is an SGI Octane SSE computer. The virtual environment system supports pick-and-release tasks. Multiple hardware devices, such as a joystick, a LogiTech™ three-dimensional mouse, and a head-tracker, are integrated to support the interactions. Audio stimuli, besides visual channels, are added to form a multi-sensory system.

The main evaluation software is designed to facilitate fast integration of existing applications. It uses OpenGL Performer-based functions and objects. Several user-friendly interfaces are also developed to help in carrying out the evaluators' operations. The system has the potential to serve as a tool with which to build additional evaluation experiments.

CONTENTS

Chapter 1 Introduction 1
  1.1 Overview and Motivation 1
  1.2 Definitions of Key Terms 7
  1.3 Problem Statement 9
    1.3.1 Tasks Recognized by Principal Investigators 9
    1.3.2 Research Objective 10
  1.4 Contributions 10
  1.5 Organization of this Thesis 12
Chapter 2 Background and Related Work 13
  2.1 Output and Input Mechanisms 14
    2.1.1 Visual Devices 14
    2.1.2 Auditory Feedback 17
    2.1.3 Input Devices 18
  2.2 Interaction Techniques 22
    2.2.1 Navigation 22
    2.2.2 Selection and Manipulation 25
    2.2.3 System Control 28
  2.3 Human Factors in Virtual Environment for Training Applications 29
  2.4 Design and Evaluation Techniques 31
Chapter 3 Our Approach 36
  3.1 Overview 36
  3.2 Environment Setup 37
    3.2.1 Image Generator 37
    3.2.2 V4® HMD and Dome 37
    3.2.3 Choice of Tracking Devices for Selection and Manipulation 43
    3.2.4 Choice of Device for Navigation 45
  3.3 Visual and Auditory Feedbacks 46
  3.4 Pick-and-Release Tasks 49
    3.4.1 Navigation Metaphor 50
    3.4.2 Selection and Manipulation Metaphor 51
Chapter 4 Implementation 52
  4.1 Libraries and Toolkits 52
    4.1.1 OpenGL Performer Graphics Library 53
    4.1.2 Motif, Buttonfly, and Gnuplot 54
  4.2 Simulator Architecture 54
  4.3 Visual Databases 56
    4.3.1 Models 56
    4.3.2 ASE Scene Generator 60
  4.4 Simulation Loop 60
  4.5 Distortion Correction 63
  4.6 Instructor / Operator Interfaces 65

4.7 Data Collection and Playback Design 66
  4.7.1 Data Collection 66
  4.7.2 Playback Design 67
Chapter 5 Results and Future Work 70
5.1 Results 70
  5.1.1 Level One 70
  5.1.2 Level Two 74
5.2 Conclusion 77
5.3 Future Work 77
References … 79
Appendix A User's Guide 88

LIST OF FIGURES

Figure2.1 Head-Mounted Display 14
Figure2.2 BOOM™ 15
Figure2.3 CAVE™ 16
Figure2.4 Elumens™ VisionDome 16
Figure2.5 ImmersaDesk™ 17
Figure2.6 HiBall 19
Figure2.7 Ascension FasTrak™ 20
Figure2.8 LogiTech™ 3D Mouse 20
Figure2.9 CyberGlove® 21
Figure2.10 Virtual Representation 25
Figure2.11 Hand-Held Miniature 27
Figure2.12 Virtual Tricorder Interface 29
Figure2.13 Effectiveness Matrix 31
Figure2.14 Three-Stage Design and Evaluation Technique 32
Figure2.15 Visual Scenario [Bowman et al. 2001] 33
Figure3.1 Dual-Head Octane® Workstation 37
Figure3.2 Projection-Dome Display 38
Figure3.3 Triuniplex Projector inside the Dome 39
Figure3.4 System Diagrams of the Two Triuniplex Projection-Dome System 41
Figure3.5 Instructor / Operator Working Area 41
Figure3.6 Elumens™ Projector-Based System 42
Figure3.7 Visual Cue 1 – Wireframe 48
Figure3.8 Visual Cue 2 - Highlight and Image 48
Figure4.1 Software Libraries and Toolkits Dependency 52
Figure4.2 From Scene Graph to Visual Display 54
Figure4.3 Simulator Architecture 55
Figure4.4 Level One Model 58
Figure4.5 Level Two Model 58
Figure4.6 Corridors 59
Figure4.7 Pseudo Code of the Simulation Loop 61
Figure4.8 Multiple View Implementation 62
Figure4.9 Distortion Correction – Create Six Faces of a Cube 63
Figure4.10 Sphere Mapping 64
Figure4.11 Truncated Sphere Mapping 64
Figure4.12 Subject Information Menu 65
Figure4.13 2D Plot of Exposure Data (Level Two) 68
Figure4.14 3D Plot of Exposure Data (Level Two) 69
Figure5.1 Initial View of the Subject 70
Figure5.2 Example of Pick-and-Release of a Red Ball (Level One) 71
Figure5.3 Example of Pick-and-Release of a Green Ball (Level Two) 74

LIST OF TABLES

Table 2.1 System Setup and Measured Factors 34
Table 3.1 Some Properties of the VR4® HMD and the Projection-Dome 42
Table 3.2 User Actions, Tracker Functions, and the Camera Control 45
Table 3.3 Joystick Functions and the Camera Control 46
Table 3.4 Visual and Auditory Feedback 47
Table 4.1 Objects and Sizes 57
Table A.1 Tokens and Specifications (1) 90
Table A.2 Tokens and Specifications (2) 93
Table A.3 Tokens and Specifications (3) 94

Chapter 1 Introduction

The screen is a window through which one sees a virtual world. The challenge is to make the world look real, act real, sound real, feel real.

[Sutherland 1965]

1.1 Overview and Motivation

Virtual reality (VR) or virtual environment (VE) is real-time interactive graphics with three-dimensional models, combined with display technologies that give the user immersion in the model world and allow direct manipulation [Bishop et al. 1992]. As to the first, the computer-generated illusion proposed by Sutherland in the 1960s has driven the human computer interfaces in the decades since. He built the first Head-Mounted Display (HMD) coupled with a tracking system [Sutherland 1968]. Since then, VR systems have advanced to offer the opportunity to immerse a user in a synthetically computer-generated environment of true three dimensions (3D). Inside this virtual world, the images are not frozen. The user can navigate and interact with wire-framed objects in real time. The software ensures that the visual scene is always appropriate to the user's head positions to create the correct perspective view of three-dimensional objects.

Virtual environment systems have three major elements: interaction, 3D graphics, and immersion [Pratt, Zyda & Kelleher 1995]. Various platforms support dynamic rendering of three-dimensional scenes at a real-time refresh rate on the high-resolution display system [Foley et al. 1990]. The evolution of effective software tools, such as for graphic rendering, data preparing, lighting and shading, allows the developer to build the environment, which could be anything from 3D objects to an abstract database.

The essence of VE is immersion. The sensation of immersion or presence elicited by immersive VE technology indicates that VE applications may differ fundamentally from those commonly associated with graphics and multimedia systems [Hodges et al. 1995]. This sensation is greatly influenced by the quality of the display, the extent of sensory information, the ease of navigation, the ability to modify the environment, and how comfortable the user feels when using a computer [Barfield & Weghorst 1993; Lewis & Griffin 1997].

We have experienced a proliferation of display devices, for example, head-mounted display (HMD) [Fisher 1989; Teitel 1990], Binocular Omni-Orientation Monitor (BOOM) [McDowall et al. 1993], projection dome [Hirose 1995], CAVE Automatic Virtual Environment (CAVE) [Cruz-Neira, Sandin & Defanti 1993] and workbench [Kruger et al. 1995]. These displays may supply the user's eyes with a stereoscopic or monoscopic view. Based on the level of presence presented by the devices, VE is categorized into immersive system, desktop or fish tank environment, and augmented reality.

The importance of display technologies lies in that the user experiences a 'first-person' view of the VE, which opens up totally new interaction modalities [Vince 1995]. The human then becomes the center of the system. In the virtual world, users can select, pick up, position, orient, and place objects, and do all interactions to simulate possible or even impossible actions in the real world. The naturalness of the interaction metaphors is greatly compromised by real-time tracking [Meyer & Applewhite 1992] and the capabilities of tactile (touch) and haptic (force) devices [Gomez & Langrana 1995; Acosta et al. 1999]. Users should be unaware of the technology while experiencing the intuitive interaction.

Audio becomes a natural feature to complement the interactive visual and tactile domains [Durlach 1991]. Current technology is already capable of supplying binaural signals that model the attenuation of pressure waves entering the user's ear canals, thus simulating the way our ears perceive sounds in the real world. This provides valuable audio cues for the brain to localize the source of sounds. Additionally, the source of sound may be incidental: sound associated with some erroneous interactions may warn us of our mistakes. The combination of visual, tactile, audio and possibly even olfactory outputs [Dinh et al. 1999] forms a multi-sensory system. They are useful features to assist in human virtual environment interaction [Meyer & Applewhite 1992; Gomez & Langrana 1995].

The unique features and flexibility of VE give it extraordinary potential for use in work-related applications. Many applications have been greatly improved during the past thirty years. They span the fields of virtual wind tunnels [Bryson & Levit 1991], flight simulation [Platt, Dahn & Amburn 1991], training [Loftin & Kenney 1995; Tate, Sibert & King 1998; Chen et al. 2001], education [Brelsford 1993; Dede, Salzman & Loftin 1996; Dean et al. 2000], architectural walkthroughs [Brooks 1986], data visualization [Bryson & Levit 1991; Lin, Loftin & Nelson 2000; Rosenthal et al. 2001], health care [Fechter et al. 1996; Moline 1997], manufacturing [Cobb, D'Cruz & Wilson 1995], and others [Cruz-Neira & Lutz 1999]. The outcome of prior research demonstrates that VEs provide significant benefits over other methods in these areas.

Among them, training applications are usually more complex. As virtual environments become more popular for training, they have appeared more frequently within the workplace. Using VE techniques in training applications has several advantages:

(1) Virtual environments provide many hardware devices and software environments, which serve as the simulators of real world operations. The simulators are quite forgiving in the way they tolerate mistakes. They are safe and cost-effective.

(2) Virtual environments allow for total control of the interaction by the trainer. For the trainee, as well, the ability to customize the virtual environment to the individual opens up a range of new possibilities. This offers the possibility of providing innovative training strategies. The systems also enable training rehearsal which is especially useful to enhance learning.

(3) Virtual environments present a unified workspace allowing more or less complete functionality without requiring that all the functions be located in the same physical space. Networked virtual environments (net-VE) or collaborative virtual environments (CVE) form the basis for a new generation of applications.

The choice of display delivery devices is very important when designing virtual environment training applications. It depends upon the goals of the particular tasks, the cost and the technical complexity developers are willing and able to assume. At Johnson Space Center (JSC) at the National Aeronautics and Space Administration (NASA), two types of delivery systems, one projection-dome based and the other HMD-based, are currently used for pre-adapting astronauts for micro-gravity. Both HMD and Dome can build a critical link between a virtual environment and the physical world. Also, they provide intuitive and natural interaction metaphors. But they are quite different in image performance (e.g., brightness, resolution, field of view, and contrast ratio), physical space limitation, weight, portability, and cost. In addition to these physical aspects, the interfaces they present also vary. Previous studies show that less-than-perfect display devices later lead to critical lapses in performance in real-world conditions [Cruz-Neira & Lutz 1999].

On the other hand, human factor issues related to VEs for training applications are not clear. These issues include human performance, potential cyber-sickness [Pausch, Crea & Conway 1992; Kennedy et al. 1993] and social effects [Kallman 1993; Stanney, Mourant & Kennedy 1998]. The lag time between the user's movement and the resulting change in the virtual display are among the factors that can induce cyber sickness [Kennedy & Stanney 1996]. It is necessary to determine how the display devices influence the users' performance and their overall effectiveness [Wilson 1997].

Most previous efforts have been expended on the development of new equipments and software rather than on the evaluation of whether VEs accomplish their stated goals. User-centered design and evaluation lag behind what is needed [Gabbard, Hix & Swan 1999]. There are a few authors who compare devices on domain-specific tasks [Chung 1992; Pausch, Shackelford & Proffitt 1993], but the comparisons are not formalized [Bowman & Hodges 1999]. To address this imbalance has been identified as one of the most pressing issues [Shneiderman 1998; Darken et al. 1998; Stanney, Mourant & Kennedy 1998].

Previous criteria used to evaluate human performance in virtual environments usually focus on task accuracy and task completion time. The testbed evaluations are generally based on a single task that subjects like to perform in a virtual environment, such as selection and manipulation [Poupyrev et al. 1997], navigation [Bowman, Koller & Hodges 1997; Bowman, Johnson & Hodges 2001] or use of the menu system [Bowman & Wingrave 2001]. But we cannot obtain the information about the intra-task performance. Some of them assume that display devices don't affect the task performance. Such summative evaluations [Hix & Hartson 1993] may provide quantitative and qualitative data for a single step, but need to be extended for real world task performance measures. Standardized methods of evaluation for real training need to be developed and systematic data gathering protocols need to be implemented.

This thesis project, therefore, addresses the development of a testbed to aid in the evaluation of two types of display delivery systems on real-world tasks. The comparison is implemented as a virtual environment system, which integrates two display devices – a customized projector-dome with Elumens™ projector and a Virtual Research's V4® HMD, tracking devices – a LogiTech™ head tracker and a 3D mouse, and a navigation device – a digital joystick. Our testbed is different from previous research in addressing the evaluation of the pick-and-release task, which is involved in most training processes. It is the combination of three common tasks: bodycentric navigation, handcentric manipulation, and handcentric selection.

1.2 Definitions of Key Terms

We define some of the general terms that relate to this work.

Virtual Environment (VE) / Virtual Reality (VR): Real-time interactive graphics with three-dimensional models, when combined with a display technology that provides the user immersion in the model world and allows direct manipulation. For the purpose of this thesis, they are used interchangeably.

Immersion/Presence: The feeling of "being there" that is experienced in some VEs. VE users are immersed when they feel that the virtual world surrounds them and has to some degree replaced the physical world as the frame of reference.

Real-time: Displayed at a frame rate that ensures that images move smoothly as the view direction changes. The minimum frame rate that is considered to be real-time might be between 10Hz and 30Hz for different applications.

Head-mounted display (HMD): A full-immersive display device worn on the head to provide a view of a computer generated scene. The screen is liquid crystal diode (LCD) or cathode ray tube (CRT) type.

Projection-dome display (Dome): A semi-immersive display device. The image is projected onto a spherical surface based on the astronomical planetarium. Usually the projector is CRT based.

Field of view (FOV): The largest solid angle where incident light appears on the image. In an HMD, the FOV is often quoted as horizontal and vertical angles.

Field of regard (FOR): The total display field of view viewable by a user with the full range of head movement. It is potentially larger than FOV.

Tracker: A device that measures 3D position, and sometimes orientation, of a subject relative to the known source. Common tracker types are electromagnetic, optical, ultrasonic, gyroscopic, and mechanical linkage.

Human-computer interaction (HCI): The exchange of information between human beings and computers during a task sequence for the purpose of controlling the computer (from the point of view of the human) or informing the user (from the point of view of the computer).

I will use the following terms in describing the experiments:

Session: A visit to the laboratory to perform several tasks during multiple VR exposures.

Exposure: One “go” in the VE system.

Task: There is typically one main task per VE exposure.

Trial: A single instance of the task.

1.3 Problem Statement

1.3.1 Tasks Recognized by Principal Investigators

In the proposed investigation by NASA [Harm 1998], the investigators addressed our task as "primary responsibility for the design, development and implementation of the visual environment data bases for linking display, tracking, and interaction devices to both VR systems". Here, "both VR systems" refer to the VR4 head-mounted display and the projector-dome.

In the proposed "experiment design and methods" part of the investigation, Harm [1998] specified that the tasks appeared similar to the one by Lampton [Lampton et al.1994]. The basic scene is as follows: "*the VE is a room with a set of 15 colored balls (orange, blue, green, white, yellow; 3 of each color) along one wall and 15 matching platforms along the opposite wall and a single column located in the center of the room. The task requires participants to move the 15 balls on the left side of the room over to the matching 15 platforms on the right side of the room. In addition, the subjects are required to move clockwise once around the center column before placing the ball on the matching platform. Participants will use a standard 3-button mouse to move about and manipulate the objects in the VE*". A standard 3-button mouse only supports two-dimensional (2D) operation and is suitable for the traditional 2D WIMP (Window, Icon, Menu, Pointing) graphic user interface (GUI). In our system, we used a LogiTech 3D mouse as a manipulation and selection tool to support 3D interactions.

In the proposed presence questionnaire (PQ), the independent variables that are related to this thesis work include the type of the VR system: HMD or dome, the length of exposure: 30min, 60min or 90min, and the session: the first, the second, or the third exposure.

1.3.2 Research Objective

Several fundamental technological problems to make the environment practical serve as research objectives. The first is related to the realism and choice of display generators and interfaces. Due to the physical differences of HMD and Dome, the interfaces they present are quite different. Multiple devices are available for interactions. Hence, we need to consider the configuration of interface devices for interaction with applications and how many variables are added.

The second objective is to investigate current interaction techniques and integrate them into the system. Many interaction metaphors create the interaction medium and allow an increase in the performance. They are usually effective in decreasing the completion time and increasing accuracy.

The third objective is to implement the software. This includes model creation, simulation loop, data collection, and the interfaces to help the investigators' operation.

1.4 Contributions

The overall goal of this research is to build a virtual environment system for qualitative and quantitative comparison of HMD and Dome. Our contributions are listed below.

- Provide a testbed platform for a summative evaluation of interaction.

The testbed is based on general tasks, i.e., pick-and-release, which possibly are executed in the training process. They are the combination of three components: bodycentric navigation, handcentric selection, and handcentric manipulation. This is more realistic than previous testbeds, which usually assess each task component separately.

- Provide a testbed for understanding subject performance.

We measure data in the form of accuracy of placement, error, task completion time, and user comfort. Traditional data recording was usually done either by videotape or written account. We implement a playback module and a data-recording module to assist in this process. During exposures, our data-recording module records all operations the subject performed as well as accuracy and completion time. The playback function can take the data as input and replay the whole trial process. This module greatly facilitates the evaluation process.

- Implement a multi-sensory environment including visual and auditory feedbacks.

Because the real world provides us much sensory input including visual, hearing, touching and so on, we hope the system simulates as many sensors as possible similar to those in the physical environment. But haptic interfaces are still weak, and they are rarely used in ground training applications. In our system, we integrate one more cue - auditory stimuli besides visual channels for assisting subject interaction.

1.5 Organization of this Thesis

This thesis is divided into five chapters and one appendix. In this chapter, *Introduction*, we have introduced the subject of a comparative study for VE training, motivated by the need for research in this area, defined the terms we use, and described our objective and contributions of this research.

Chapter two, *Background and Related Work*, presents a detailed look at previous work that has influenced ours or provided background to the current research. This includes: output and input devices, current user interaction techniques, human factor issues, and virtual environment design and evaluation techniques.

Chapter three, *Our Approach*, discusses the ideas of our design with emphasis on the environment setup, visual database creation, visual and audio feedback design, and task analysis.

Chapter four, *Implementation*, provides the software architecture with all of its component parts.

We conclude in Chapter five, *Results and Future Work*, with a discussion of the results of this work and possible future work. Finally, Appendix A presents a simple user's guide.

Chapter 2 Background and Related Work

All virtual environments are “through the window” systems. Visual feedback is without question the most dominant channel in the overall VR system. Nonetheless, no “one size fits all” display exists for all applications [van Dam et al. 2000]. Display devices have appeared in many forms. The advantages and disadvantages of these devices are strongly task-related. Thus, mapping appropriate display characteristics to tasks is essential for the development of VEs. Moreover, evaluating display devices is equivalent to assessing how well the devices match the representative tasks with regard to device effectiveness.

To perform a task, a variety of interaction techniques can be used to associate the user’s actions captured by input devices with the resulting actions by output devices. Both the interaction techniques and the devices have advantages and disadvantages with regards to ergonomics. Therefore, it is important to understand the human factors of these techniques. In order to address a comparative study of HMD and Dome delivery systems, several areas of prior research must be reviewed: output and input mechanisms, interaction techniques, human factor issues, and virtual environment design and evaluation techniques.

2.1 Output and Input Mechanisms

2.1.1 Visual Devices

HMDs are the most commonly used display devices that provide a full-immersive environment. These displays present stereoscopic or binocular images on the small displays located near the head with appropriate lenses (Figure 2.1). If coupled with a head-tracker, the HMD system is called a head-coupled system. If further integrated with a joystick, the head-coupled system allows the user to visualize data spatially by real-time tracking and joystick-navigation. HMDs are portable and proper for some applications where the user works in isolation or needs to look around in a VE [Lantz et al. 1996]. The shortcoming, however, is their low resolution. This leads to problems of alias or spatial display artifacts [Edgar & Bex 1995; Robinett & Rolland 1992]. Compared to shutter glasses and counterbalanced displays, HMDs are the heaviest of all. Thus, it is not surprising that user fatigue is associated with prolonged use of HMDs.

Figure 2.1 Head-Mounted Display

See-through HMDs, currently used mainly for augmented reality, let the user view computer-generated images that are superimposed over the real world. Such displays are typically integrated into existing work practices and procedures, providing real-time supplemental information to the user [Rosenthal et al. 2001]. In augmented reality applications, accurate registration is hard between the real and the virtual objects [Azuma & Bishop 1994]. For the purpose of this thesis, HMD refers to the non see-through type.

The Fakespace binocular omni-orientation monitor (BOOM™) and BOOM3C™ overcome the encumbrance and low resolution of HMDs. BOOM™ is supported with a balanced arm whose joints contain mechanical trackers that enable position detection. The user does not "wear" it, but at least one hand must hold the arms (Figure 2.2). Therefore, those applications that need two-hand interaction do not fit with the BOOM™. The advantage of the BOOM is that it can be seamlessly integrated into the user's work activity, exploiting the work habits that the users already have in place [McDowall et al. 1993].

Figure 2.2 BOOM™

Spatial immersive displays (SIDs) are a second class of display types providing semi-immersive VEs. They physically surround the viewer with a high-resolution panorama of imagery produced by video projectors. Two most common types are CAVEs and Domes. The EVL CAVE™ projects images onto a four-wall plus the floor to form a cube with five projectors (Figure 2.3). By contrast, images in domes can be projected onto a curved surface with one or more projectors. An example is the Elumens™ VisionDome (Figure 2.4) [Http1]. It uses only one projector to present images on a spherical surface.

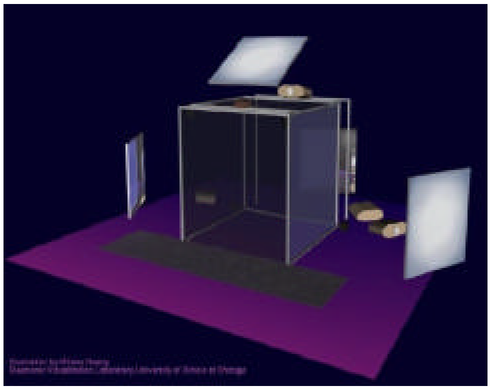


Figure 2.3 CAVE™

Figure 2.4 Elumens™ VisionDome

In a SIGGRAPH 1996 panel [Lantz et al. 1996], the panelists illustrated that the dome displays can provide a better sense of presence than any other displays by projecting onto a spherical surface. Users feel comfortable because the topology of the projection is consistent with the real-world one.

SIDs provide a balance between immersion and spatial object visualization. Rather than being fully isolated from the physical world, users still can see their own bodies and other surroundings. In most cases, they wear a pair of shutter glasses for stereo viewing. Therefore, users might have a good sense of self due to the subtle cues provided by the physical environment. This property makes SIDs well suited for spatially rich applications such as environmental walk-through and flight simulations [Lantz et al. 1996]. Likewise, the body-centric judgment about 3D spatial relations is much easier [Mine, Brooks & Sequin 1997; Pausch, Proffitt & Williams 1997]. The shortcoming of SIDs, however, is that the seams across display surfaces can cause visual discontinuities. Sophisticated image blending algorithms are generally required for making images appear seamless [Raskar et al. 1999]. Another practical drawback of the CAVE™ and the Dome is their cost and the room space they need to accommodate the system.

Virtual Model Displays (VMDs) are capable of generating a 3D virtual world where the effect lies in the volume of space, a space that that is equivalent to inside and outside the display surface. Like SIDs, VMDs also provide high-resolution and can be shared by several users. Users wear a pair of shutter glasses for stereo viewing as well. Figure 2.5 illustrates a user who is interacting with data on an ImmersaDesk™ using the ImmersaDesk™ Wand. Particularly, VMDs are well suitable for providing exocentric views of virtual models such as those used in medical data visualization [Rosenthal et al. 2001] and geosciences visualization [Lin, Loftin & Nelson 1999].

Figure 2.5 ImmersaDesk™

2.1.2 Auditory Feedback

Auditory feedback [Begault 1994] is powerful when successfully combined with visual feedbacks. Experiments have shown that accurate acoustic modeling gives users a strong sense of presence in a virtual environment [Tsingos et al. 2001]. Auditory research has examined the effectiveness of the use of localization, which represents the highly realistic, 3D localized sound. An advantage is its usefulness for reinforcing perceived visual quality and serving as sensory substitution when no haptic tactile feedback is present [LaViola 2000]. Besides 3D sound, the 2D sound called sonification addresses turning information into meaningful sounds. For example, we can simulate collision detection noises, object dropping, and thunderstorm.

2.1.3 Input Devices

Input devices monitor the user's actions and feedback signals that reflect the status of interaction in a virtual environment. Although they have appeared in numerous forms, each has its unique contribution to user interactions. The pros and cons depend on the applications and are strongly task related.

The position of the user is most fundamental information that we must know. It is often given in terms of location (x, y, z) and orientation (heading, pitch, roll). In many applications, six DOF trackers are used for every positioning need. They can be placed on HMDs, gloves, body joints, and in hand-held devices. The main products on the market include mechanical, optical, magnetic, and acoustic trackers.

An example of a mechanical tracker is the Fakespace BOOM™. It uses booms and angle encoders to determine position. It is best used in single user applications that require limited volume restricted by the mechanical linkage [Applewhite 1991]. The mechanical trackers are the most precise and fastest trackers.

Optical trackers make use of a number of small targets on the body, either flashing infrared LED's or small infrared-reflecting dots. Two or more cameras surround the subject and pick out the markers in their visual field. Software correlates the marker positions in the multiple viewpoints and uses the different lens perspectives to calculate a 3D coordinate for each marker. Optical trackers commonly report high data rates. An example is Ascension's laserBIRD® with measurement rates of 240Hz [Http2]. The limitation, however, is the processing time that is needed to analyze the several camera images and determine each target's position. The HiBall Tracking system [Welch et al. 2001] can achieve very high accuracy with 2000Hz rates and low latency. But its big size restricts its usage except in head tracking. Hand tracking is indeed rarely used although it does exist (Figure 2.6) [van Dam et al. 2001].

Figure 2.6 HiBall
HiBall is used as a head tracker or a *3D digitization probe*: A pencil-shaped fiberglass wand is linked to the end of a HiBall

Magnetic trackers are robust and widely used for head and hand tracking. They are relatively small, lightweight, and cheap. A source element radiates a magnetic field and a small sensor reports its position and orientation with respect to the source. An example is the Ascension FasTrak™ (Figure 2.7). It is very frequently used coupled with the V4® HMD system in our lab. The small cube is used as the head tracker, and the stylus is used as the hand tracker. One obvious drawback of magnetic trackers is that they are restricted to environment without heavy metallic objects that might distort the magnetic field [Applewhite 1991].

Acoustic trackers use the time of flight of high-frequency sound pulses from an array of sources to an array of receivers to determine the position and orientation within the working area. Precise placement of the microphones allows the system to locate the source in space to within a few millimeters. If multiple acoustic trackers are used together, they can operate at non-conflicting frequencies. The disadvantage of acoustic trackers is that they are sensitive to ambient acoustic noise. An example is the LogiTech™ 3D mouse (Figure 2.8). It has the benefit of a suitable number of constraints. Users usually feel comfortable to use it with six DOF motion interactions and buttons.

Figure 2.7 Ascension FasTrak™

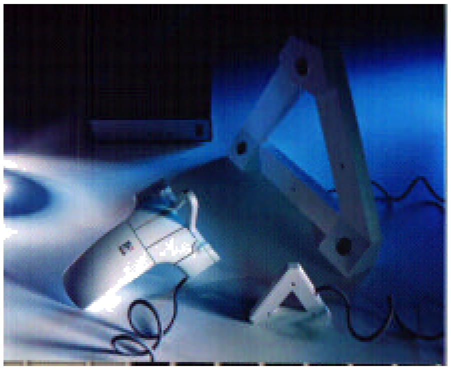


Figure 2.8 LogiTech™ 3D Mouse

Unlike other devices that serve as either input or output, the haptic and tactile devices provide both output and input. This means that they provide a closely coupled, bi-directional flow of information between users and the virtual world [Richard et al. 1996].

Thus, users simultaneously feel forces reacting with their bodies as they exert forces with their bodies.

Datagloves are one type of haptic and tactile devices. They utilize gesture recognition techniques for computer output and support many degrees of freedom. Hand movement is captured and data are produced in response to the user's actions. Small pads along the fingers in glove are activated to simulate the touch condition when a collision is detected between the user's virtual hand and a virtual object. For example, the CyberGlove® (Figure 2.9), probably one of the best gloves on the market, measures finger joint flexion, and the 22-sensor model also measures the angle between adjacent fingers, the extent of thumb crossover, palm arch, wrist flexion and wrist abduction. The dataglove, and other haptic and tactile devices, are particularly suitable for those applications that require high realism or a high DOF in the simulated interaction. The problem with datagloves is that they cannot be easily used with too few constraints and too many possible configurations. The most compelling applications of haptic and tactile devices today are in medical fields [Rosenthal et al. 2001; van Dam et al. 2000].

Figure 2.9 CyberGlove®

Speech input is also an input mechanism useful for interaction. Especially when both hands are occupied, speech input is a valuable tool for issuing commands. The disadvantage however is that it is very hard to have the system differentiate between a speech command and general conversation. This problem can be partly solved by a push-to-talk scheme [LaViola 2000] with the assistance of certain devices. But building a sophisticated natural language recognition system is still a challenge.

2.2 Interaction Techniques

The potential user task space involved in VE applications is enormous. A thoughtful approach to understanding the tasks is by understanding smaller, yet representative subtasks. In fact, taxonomy has classified interaction tasks into a few uniformed interaction components: navigation, selection, manipulation, and system control [Mine 1995; Bowman & Hodges 1999]. An alternative classification is based on *frames of reference* [Howard 1993]. Tasks are classified into egocentric frames (those that are defined with respect to some part of the observer, e.g., handcentric, headcentric or bodycentric) and exocentric frames (those that are defined with respect to points external to the observer). Navigation, selection, and manipulation can be implemented in either the exocentric or egocentric way when integrated into different applications and when performing different tasks.

2.2.1 Navigation

Navigation provides a way to move the user's viewpoints and body towards specific places in a virtual world. It has been divided into two subtasks: wayfinding and travel. Wayfinding, a cognitive component, is the process of acquiring spatial knowledge and awareness of the surrounding space; travel, a motor component, manages the process of moving viewpoint from place to place [Hinckley et al. 1994].

Navigation metaphors are generally applied to minimize the user's loading and improve task performance during VE exposure. Flying vehicle control [Ware & Osborne 1990; Slater, Usoh & Steed 1995] is a very common metaphor. This interaction relies on hand-based gestures and the orientation of hand-held pointing devices to determine direction and velocity. The devices can be joysticks, datagloves, and wands. It is good for imprecise but fast long-range navigation. One-to-one or highly exaggerated factors can be used to register the displacement of the physical device to the virtual camera movement.

Some metaphors have been implemented that are similar to the counterpart problem in every day life. For example, Darken and Sibert [1993] applied a virtual map in their toolkits to obtain orientation information, and thus to increase the user's spatial awareness. The scaled world in the hand provides an exocentric view of the complete virtual world. A similar technique is the scene-in-hand metaphor [Ware & Osborne 1990]. It allows the user to manipulate virtual objects directly through interaction with the mini-world. These metaphors are well suited for precise short-range navigation.

Some metaphors [Darken, Cockayne & Carmein 1997; Slater, Steed & Usoh 1993] employ the user's physical movement to do navigation; for example, a user who is standing on a treadmill can move toward any directions in the physical world. A small distance the user moved physically is mapped to a large one in the virtual world. In this environment, the user is usually fully immersed and performs actions in the working volume of the tracking system. Usoh and coworkers [1999] found that real walking was much better as a mode of locomotion and projects higher presence according to an experimental study to compare real walking with virtual walking and the push-button-fly metaphor.

Other metaphors, for example, gaze-based navigation [Mine 1997] and flying into hand-held miniature [Pausch, Brockway & Weiblen 1995], are also popular. Mine's method [1997] determines the direction of travel by the user's head orientation and the direction of fly by the vector between the user's two hands. The speed is proportional to the user's hand separation. Pausch, Brockway, and Weiblen's method [1995] brings the user to an updated full-scale environment by pointing to the destination. The drawback is that users tend to lose their direction since the resulting image is immediately displayed.

If travel is used as a service of some other tasks rather than an end unto itself, we can employ target-based travel techniques. In the method presented by Bowman and coworkers [1998], a map is drawn in the virtual environment, and the user holds a stylus in hand, dragging its virtual body prop on the map to the destination. When the virtual prop is dropped, the user flies to the new location in a smooth animation.

For input devices to provide almost exhaustive control over the virtual world, the DOF required in the virtual world must be addressed [Zhai & Senders 1997]. When implementing navigation through a complex environment, the number of degrees of freedom under subject control is one of the most important factors. Too few or too many may result in difficulty in reaching the desired destination or losing control of the motion and disorientation.

2.2.2 Selection and Manipulation

In the virtual world, we must have virtual ‘touch’ before selection and manipulation. The user who is immersed within a VE has a natural inquisitive temptation to reach out and touch virtual objects [Vince 1995]. Including a virtual representation is a classic way to address this problem [Hinckley et al. 1994; Wloka & Greenfield 1995; Poupyrev, Tomokazu & Weghorst 1998; Pierce, Stearns & Pausch 1999]. The virtual representation is spatially registered with a physical input device which provides sensory feedback. The sensory feedback could be visual, tactile or haptic. For example, a virtual hand is commonly used to mimic touch by intersecting the virtual hand with a virtual object (Figure 2.10).

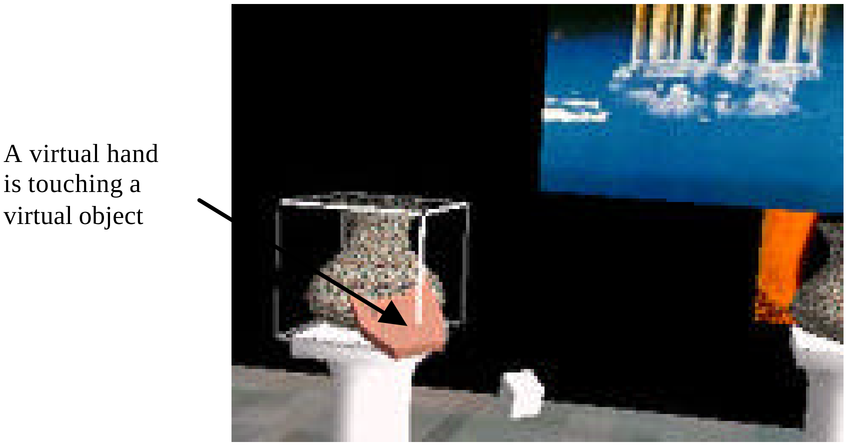


Figure 2.10 Virtual Representation

The “real touch” is a classical *virtual hand metaphor* [Poupyrev et al. 1998] to do selection by natural physical motion [Robinett & Holloway 1992]. Users accomplish tasks by traveling to a place near the object, reaching out a hand, touching and selecting the object, then manipulating it. The mapping between the virtual representation and the real hand position is based on a linear function. It is natural and close to the real world maneuvers. The drawback, however, is that the user might have difficulty to select if large objects block the user's view, or to pick up if the object is too small.

Anther type of virtual hand metaphor is Go-Go [Poupyrev et al. 1996]. The Go-Go technique uses a linear mapping if the user's hand is within a physical range or a non-linear mapping if the physical hand goes beyond. The virtual arm 'grows' to pick up the object. Since the range of selection is defined by the mapping function, Go-Go benefits remote object selection. After selection, the object is attached to the virtual hand. Users can manipulate the object by one DOF (rotation around the hand axis).

Ray casting is another metaphor for selection and manipulation when the ray is a vector or a conic volume. Any objects interacting with the ray can be selected and manipulated. This type of metaphor is distinguished by the definition of the ray direction, the shape of the ray, and methods of disambiguating the object the user likes to select. Every metaphor improves prior techniques while resulting in new constraints [Poupyrev et al. 1998].

The silk cursor [Zhai, Buxton & Milgram 1994] selects objects that fall within a transparent cubic cursor. This cursor provides a simple occlusion cue to indicate the context of what is behind, within, and in front of the cursor. The flashlight [Liang & Green 1994] uses a spotlight to ease selection of small objects. Forsberg, Herndon, and Zeleznik [1996] used a conic pointer whose direction is defined by the vector from the user's head to a hand sensor. Bringing the hand sensor closer or farther away controls the size of the cone. HOMER (which stands for Hand-centered Object Manipulation Extending Ray-casting) [Bowman & Hodges 1997] improves the Go-Go technique by allowing selection at most distances and manipulation in two DOFs (movement and rotation around the ray axis).

Other than casting a ray directly onto the life-size object, the user can shoot a ray to the object on an image plane. The image plane provides an exocentric two-dimensional view of the virtual world [Pierce et al. 1997; Mine, Brooks & Sequin 1997]. When the selection is made, the user is scaled up and the world is scaled down. Thus, the user can access objects located at a certain distance. When release is made, the opposite actions are done in reverse: the object is re-attached to the world and the user is scaled uniformly by the reciprocal of the scaling factor.

World In Miniature (WIM) metaphors [Stoakley, Conway & Pausch 1995; Pausch, Brockway & Weiblen 1995] extend the 2D image plane technique to 3D. A small version of the virtual world serves as a second viewport onto the VE (Figure 2.11). The image is

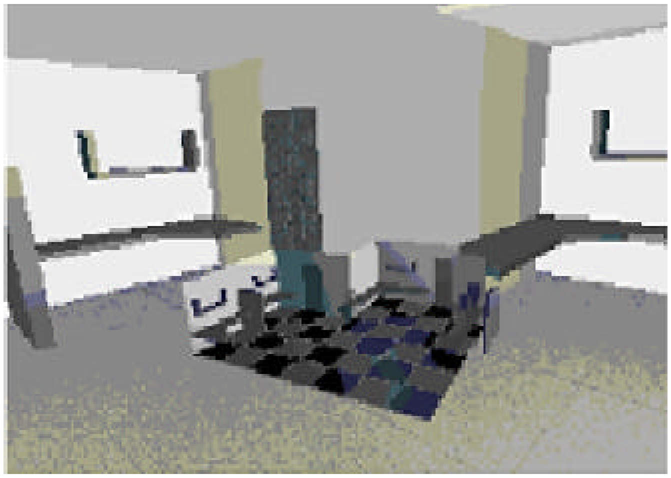


Figure 2.11 Hand-Held Miniature

presented on a head-coupled system. Both hands are tracked with Polhumus trackers. The subject can do indirect manipulation and selection through a miniature model or do direct ones through the life-size world. Recently, the WIM technique was extended to "voodoo doll" techniques [Pierce, Stearns & Pausch 1999]. The user creates his/her miniature parts (dolls) of the environment and may use two hands to manipulate these doll objects relative to each other.

As methods for navigation and manipulation, the scene-in-hand, map, and WIM metaphors work well for the virtual world including reasonably compact objects. However, when the environment is a landscape or enclosed interior space, picking up an object or moving toward where the user points will be very hard. There also exists a problem in selecting the center of the exocentric view [Ware & Osborne 1990].

2.2.3 System Control

System control is used to handle the program states or parameters which are generally difficult to be specified by direct interaction [Bowman et al. 2001], for example, changing the state of the application and the mode of interaction or parameters. Menu interaction is a general way to implement system control [van Dam et al. 2000]. It is a mode of incorporating menus and toolboxes into the virtual environment. Wloka and Greenfield [1995] designed a virtual tricorder interface to do menu selection (Figure 2.12). Bowman and Wingrave [2001] implemented a glove interface which allows the user to interact with the virtual environment using gesture commands or menus "floating" in space. The menu can be one-dimensional [Liang & Green 1994; Mine 1997], two-dimensional [Lin et al. 2001], or three-dimensional [Jacoby & Ellis 1992; Conner et al. 1992; Bowman & Wingrave 2001].

Figure 2.12 Virtual Tricorder Interface
The user is using a virtual tricorder to select items from a virtual menu. The physical prop and the virtual tricorder have a familiar transition which reduces the user's cognitive load.

2.3 Human Factors in Virtual Environment for Training Applications

Human performance efficiency is one of the human factor issues [Stanney, Mourant & Kennedy 1998]. Kim, Tendick, and Stark [1993] investigated the benefits of stereo cues in a virtual pick-and-place task. They found that performance with monoscopic display was just as effective as with the stereoscopic display when visual perspective display enhancements were provided. They further suggested that for simple tele-manipulation tasks, monocular depth cues and cognitive cues might be sufficient for effective performance. If the tasks become more complex, and the monocular and cognitive cues (accumulated through learning and past experiences) were insufficient, stereoscopic cues would enhance performance. This work also explained the reason why

Kozak and coworkers [1993] stated that there was no effective training transfer on pick-and-placement tasks, as compared to the real world and no training conditions. Many cognitive cues existed in their tested environment. Recent studies [Davis & Hodges 1995; Barfield, Hendrix & Bystrom 1997] also found that stereo displays benefited in enhancing perception and task performance in some conditions, for example, in conditions when information is presented in an egocentric view and user tasks are highly spatial (perception of shape and depth, precise placement of object, docking and visual searching) [Henry & Furness 1993; Durlach et al. 2000].

Lampton and coworkers [1994] described the Virtual Environment Performance Battery (VEPAB) developed for testing training applications of virtual environment. The goal was to transfer training from VE practice to real-world performance cost-effectively. The battery includes the determination of visual acuity, locomotion ability, object manipulation ability, tracking ability, and reaction time while viewing virtual worlds. A task called bin which is similar to the pick-and-place task was tested. Their system is comprehensive and depends on very broad variables. Thus, task performance may be limited by the users' inability to function operationally in the virtual world.

In a virtual environment, end-to-end system delay is a basic problem [Azuma & Bishop 1995; Brooks 1999]. Latency arises in tracker delays, hardware and software delays, and graphics pipeline. If it is longer than 35ms, subjects may suffer cybersickness followed by task performance drop [Kennedy et al. 1993]. To avoid this problem, we should reduce the latency below the cybersickness threshold [Ware & Balakrishman 1994; van Dam et al. 2000].

Health and safety are also essential to the success of VE technology. They have been well reviewed by Stanney, Mourant, and Kennedy [1998]. In our research, the subject has a relative long exposure time, 30 minutes and 60 minutes. Eyestrain is possible because of flicker, glare, and other visual distortions occurring in the visual display devices.

2.4 Design and Evaluation Techniques

Only in recent years have researchers paid attention to design and evaluation techniques. The literature contains a few examples of quantified user studies in these areas, although Brooks [1988] stated that the traditional uninformed and untested intuition design method is usually wrong in designing a system. Several researchers further emphasized the importance of a formalized design method and of systematical truths to evaluate the strengths and weaknesses of a design [Gabbard, Hix & Swan 1999; Kaur, Maiden & Sutcliffe 1996; Stanney, Mourant, & Kennedy 1998; Bowman & Hodges 1999].

Bowman [2001] expressed the effectiveness metric by three factors: system performance, interface performance, and user performance (Figure 2.13). Related

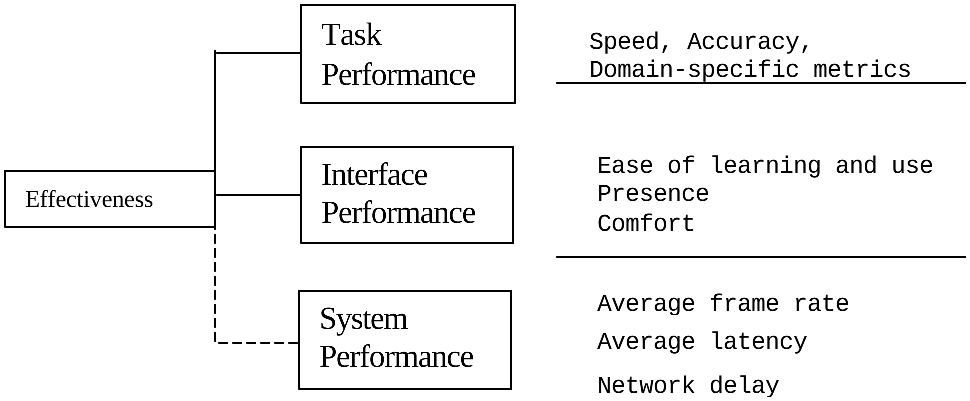


Figure 2.13 Effectiveness Matrix

parameters were also given. Among them, performance metrics, which generally lead to usability, are the most dominant factors in determining the overall effectiveness. System performance is only indirectly related to effectiveness which is marked with a dotted line in the figure. Outside factors, such as the number and color of the objects in the scene and the spatial ability of user, may also affect user performance [Bowman 2001].

Gabbard, Hix, and Swan [1999] formalized the evaluation technique as a three-phase framework of usability engineering (Figure 2.14). The first two phases require multiple

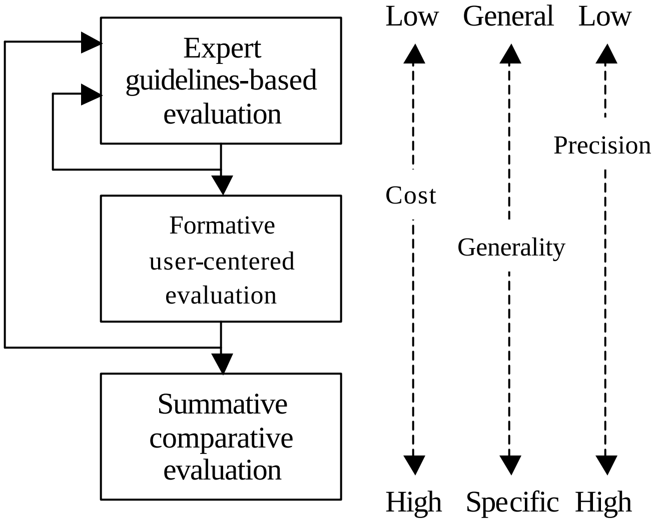


Figure 2.14 Three-Stage Design and Evaluation Technique

design iterations since the designed VE might not be suitable for the applications. They need to be further tuned. The third phase evaluation is appropriate for comparison studies with high precision. It is used for specific applications with high cost [Hix & Hartson 1993]. This method is distinguished by developing 3D user interfaces using a structured, user-centered design approach [van Dam et al. 2000]. Swartz and coworkers [1999] used the first two phases for a usability study of the CAVE application. Hix and coworkers [1999] applied the three-phase framework to real-world battlefield visualization.

Few formalized evaluations have been conducted regarding basic task components. Existing evaluations are usually implemented with a set of predefined user tasks and an environment. The goal is to measure the user's performance and obtain quantitative or qualitative results. The most closely related work has been conducted by Bowman and coworkers [2001]. They compared a Virtual Research V8® HMD with a CAVE display for a "search" task. Their goal was to build the mapping between an application and a display. The visual database they used is a corridor scene (Figure 2.15). Table 2.1 summarizes their design from task to system setup. From this table, we see that the comparison process introduces many variables that exist in different system setups, toolkits, and navigation metaphors. However, these differences may significantly affect the comparison. Completion time is the only dependent variable to be measured. Finally,

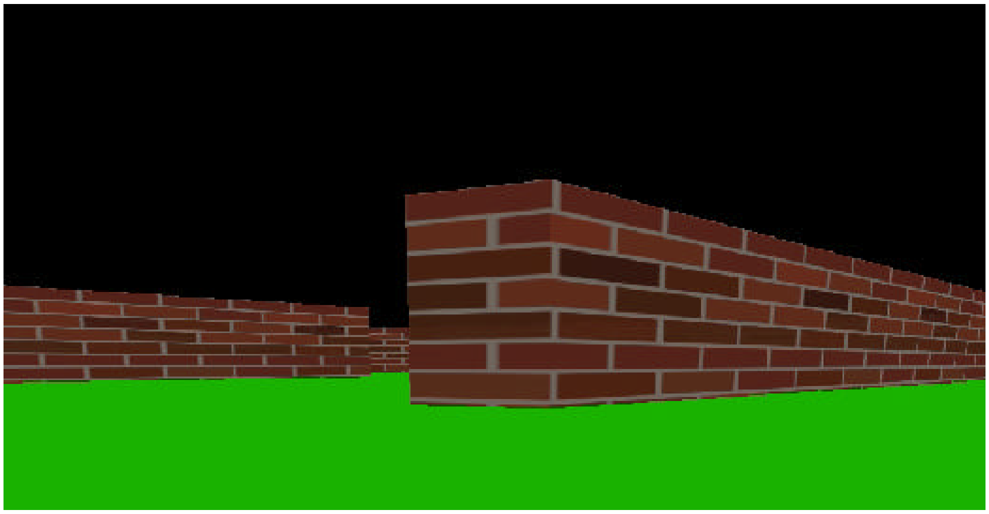


Figure 2.15 Visual Scenario [Bowman et al. 2001]

the visual database provides too little visual feedback. The user gets easily lost in such an environment.

	Task	Navigation Mode	Display Mode	Tracker
HMD	Search	Natural rotation, Manual movement (Joystick)	Binocular	Intersense IS-900 VET (Head) Intersense Wand (Hand)
CAVE		Manual rotation and movement (Joystick)	Stereo	
	System	**Toolkits**	**Visual Cues**	**Dependent Variable**
HMD	Windows NT	SVE [Kessler, Bowman & Hodges 2000]	Texture; Perspective view	Completion time
CAVE	SGI Onyx	DIVERSE [Arsenault & Kelso 2002]		

Table 2.1 System Setup and Measured Factors

Poupyrev and coworkers [1997] set up a flexible Virtual Environment Manipulation Assessment Testbed (VRMAT) for the evaluation of manipulation techniques. After recognizing a basic set of test metaphors, they examined several parameters, including distance, size, position, and occlusion in relation to a set of performance parameters, including accuracy, error rate, ease of use, and learning, and sense of presence.

Pausch, Proffitt, and Williams [1997] presented a system for comparing head-mounted and stationary displays with identical properties (resolution, field-of-view, etc.). The task is to *search* twenty targets located in a synthetic room. In their system, they attempted to avoid differing system setups that may affect the user's performance. They used the head-mounted display as a stationary display device. In this setup, the user sits on a chair and holds a tracker in his/her hand to simulate the stationary display environment. They concluded that users' search performance decreased by roughly half when they changed from a stationary display to a head-mounted display. In addition, the subject who wore an HMD reduced task completion time by 23% in later trials with the stationary display.

Chapter 3 Our Approach

3.1 Overview

In order to evaluate the VR4® HMD and the Dome display devices qualitatively and quantitatively, a virtual environment system was constructed with display devices, a joystick, a speaker, a LogiTech™ 3D mouse, and a LogiTech™ 3D head-tracker. The host computer or image generator is a customized SGI Octane workstation. Inside the Dome is an Elumens® projector and a motion base.

In our experiment, the subject uses two-hand input to perform pick-and-release tasks. The two hands assume different roles: The joystick is attached to the user's left hand to assist in travel; the 3D mouse is attached to the right hand for pick-and-release tasks. The head tracker is attached to the user's head for wayfinding. A flying vehicle control navigation metaphor and a classical virtual hand selection and manipulation metaphor are employed.

Our software is composed of seven modules. A simulation loop, the kernel of the software, includes OpenGL Performer™ based functions and objects. It is in charge of the rendering process and communicates with other modules. An Instructor/Operator Interface (IOI) provides a virtual platform to assist the operator's control over the exposure process. Finally, the evaluation process is augmented with the data recording and playback modules.

3.2 Environment Setup

3.2.1 Image Generator

Images are rendered on a customized dual-head Silicon Graphics Octane® workstation (Figure 3.1). It has two video boards so that it can support a two channel setup. This architecture contains two processors and the *uniform memory access* system [SGI]. The built-in audio of the Octane® is capable of providing auditory feedback. Most importantly, OpenGL Performer™, the library used in our software, is well tuned for this workstation so that high performance and real-time rendering can be obtained. The Octane® workstation supports two serial ports, but since we need three serial lines to control three physical devices: a joystick, a head-tracker, and a 3D mouse, we customized the Octane® by adding an additional board to support up to eight serial connections.

Figure 3.1 Dual-Head Octane® Workstation

3.2.2 V4® HMD and Dome

The V4® HMD is a lightweight (0.935kg), rugged head-mounted display. Its 1.3" active matrix LCD supports a resolution of 640x480 and has adjustments for RGB gain, hue, contrast, and brightness. The field of view (FOV) is 60^{o} diagonal at 100 % overlap.

Its optical elements are mode of multi-element glass with fully color-corrected design. The system supports an inter-pupillary distance (IPD) range of 52mm to 74mm and eye relief of 10-30 mm adjustable.

The inner surface of the dome is 3.7m in diameter, painted white, and serves as a projection surface (Figure 3.2). A gear-driven motion base rides on a 0.092m diameter bearing inside the dome. It allows computer-controlled rotation of the trainee and the projector with angular velocities up to 120^o/s and accelerations up to $200^o/s^2$ about the vertical axis. In our system, we do not use this motion base in the interest of minimizing the difference between the HMD and the Dome environments. The subject is seated upright inside the dome in both systems.

Figure 3.2 Projection-Dome Display

The problem of the Dome lies in the two Triuniplex projectors originally installed in the system (Figure 3.3). One limitation of the projectors is their low image quality in

Figure 3.3 Triuniplex Projectors inside the Dome

contrast ratio, resolution, and brightness. Another limitation is that the peripheral system to support the projectors is very complicated. Two additional monitors and video splitting boxes have to be used in order to split the visual signals in two (Figure 3.4). One is sent to the instructor or operator platform; the other is sent to the projectors. The operator can then monitor the subject's behavior during trial by viewing the same scenario as the subject (Figure 3.5). If the Octane® workstation is used, both its channels are needed to link to the Triuniplex projectors (Figure 3.4). Since both channels are occupied, other than rendering images to front buffer directly, our software has to render the undistorted images into the *back buffer*, then handle image warping to create the distorted images. These images are finally drawn to the *front buffer*. Unfortunately, the implementation is very complicated since the Performer™ library does not support back buffer rendering. We have to call lower level OpenGL functions in order to display the correct images.

Subsequently, we removed the Triuniplex projectors and switched to the Elumens™ projector. The system setup is significantly simplified (Figure 3.6). The image qualities are also greatly improved. The field of view of the visual channel presented by the Elumens® projector is 180^ox180^o. The resolution can reach 1024x1024 with qualified brightness if we render the images on a powerful machine. The Spherical Projection of Images (SPI) application interface can be integrated with Performer functions for displaying onto the spherical surface.

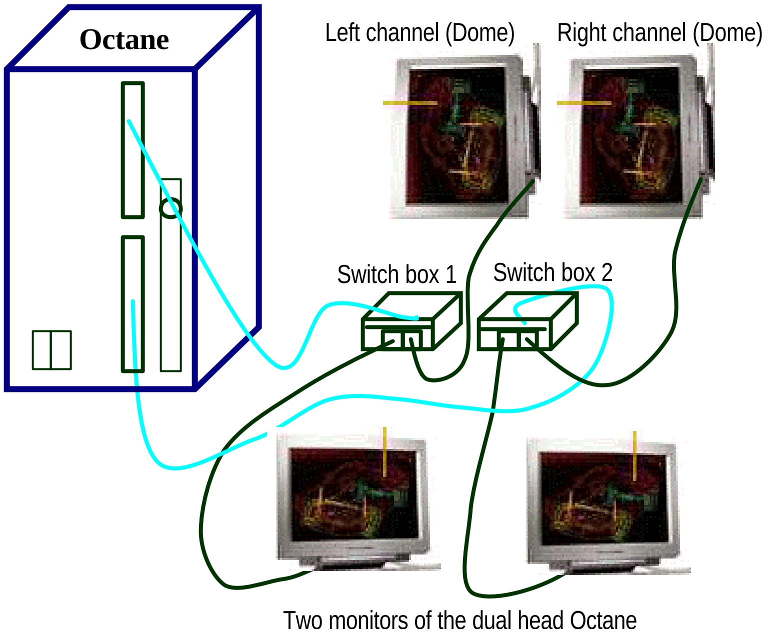


Figure 3.4 System Diagrams of the Two Triuniplex Projection-Dome System

Figure 3.5 Instructor / Operator Working Area
The operator monitors the subject's view by observing the upper two monitors which display the same images as the two Triuniplex projectors inside the dome, one for each eye. The lower monitor outputs from the SGI Octane

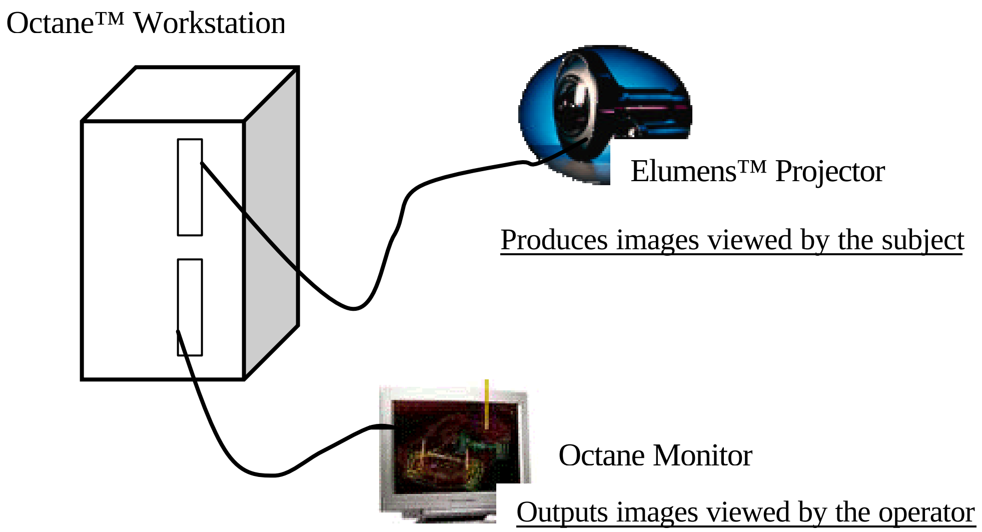


Figure 3.6 Elumens™ Projector-Based System
The operator can monitor the subject's view without adding additional hardware.

Before we describe other system design considerations, it is worthwhile to consider the immersion and visualization issues regarding the VR4® and the Dome systems. We summarize these issues in Table 3.1. Some of them have been mentioned previously. Others are listed below:

	Immersion Issues				
	FOV	FOR	Virtual Representation	Intrusion	Level
VR4® HMD	60^o	360^o	Virtual Hand	Full	Full
Dome with the Elumens® projector	180^o	180^o	Virtual Hand	None	Semi

	Visualization Issues		
	Resolution	Look Around	Program Refinement
VR4® HMD	640x480	Full	Fix location and rotation
Dome with the Elumens® projector	1280x1024	Limited	Fix location

Table 3.1 Some Properties of the VR4® HMD and the Projection-Dome

(1) *Virtual representation*, especially the hand, can give the user a strong sense of presence in a virtual environment. It is needed in both systems for pick-and-release tasks. For performing these tasks in our system, a virtual hand is employed to register explicitly with the user's hand position. It simulates the user's action in six degrees-of-freedom.

(2) *Intrusion* indicates the severity of the interface about its restriction of the senses [Cruz-Neira et al. 1992]. The HMD is highly intrusive in that it isolates the user completely from the real world. Conversely, the Dome system is not intrusive. The subject can still see the physical objects inside the Dome during the exposure.

(3) The user immersed in the head-coupled system gets a better sense of "place" and knows what is behind with a simulated 360^o FOR. In opposition, the Dome supports limited FOR, and natural look-around is not permitted without the assistance of the hand-held devices. Thus, it is harder for the user to perform egocentric wayfinding tasks in the Dome system than in the HMD system.

(4) Progressive refinement is the ability to increase dynamically the computational expense of a model during a pause in the user's response [Brooks et al. 1990]. The standard scheme is to change the level of detail in the model resolution. The HMD interface requires the user remain absolutely still to refine the display. The Dome interface requires only that the user's location remains fixed since the user is generally not head-tracked. Panning around won't affect refinement.

3.2.3 Choice of Tracking Devices for Selection and Manipulation

Choosing suitable trackers depends on the tasks to be implemented [Meyer & Applewhite 1992]. In our case, tasks are pick-and-release tasks. Head tracking is preferred for the HMD system since it can maintain an appropriate sense of presence and has a positive effect on task accuracy of manipulation and selection [Pausch, Shackelford & Proffitt 1993; Barfield, Hendrix & Bystrom 1997]. It is also helpful in modeling the egocentric points of view. Besides the head tracker, a hand tracker is required for performing pick-and-release tasks.

We have integrated an ultra-sonic LogiTech™ 3D head-tracker and a LogiTech™ 3D mouse into our system, for three reasons. Firstly, FasTrack®, a magnetic tracker, is commonly coupled with the VR4® without noticeable lag. However, it is very sensitive to metal material which may result in data distortion. It cannot be used due to the metal objects inside the Dome. Secondly, the subject's roaming area is limited around the chair area and no occlusion exists. The tracker works in the range of an approximately 5-feet long 100^o cone with millimeter precision. It also has the capabilities of updating the position and orientation at 50Hz and supporting six DOF. This is enough for our application. Finally, the price for optical trackers is usually high. The LogiTech trackers are relatively cheap.

The 3D mouse is employed to measure the user's hand position in both the HMD and the Dome system. As mentioned previously, the head tracker was attached to the user's head only in the HMD delivery system since head tracking was not necessary in the Dome system. Table 3.2 lists trackers' functionalities and the corresponding system and camera controls. The middle button of the 3D mouse is used to switch between the states of pick and release operations. The object is attached to the virtual hand if the object has not been picked up, or is dropped to the current position if the object has been picked up.

Once the object is attached to the virtual hand, its relative position to the hand will not be changed before being released. All six degrees-of-freedom of the mouse are counted in our system. Here we determined that the delays of the tracking systems for measuring the subject's performance can be ignored, based on previous studies [Ware & Balakrishnan 1994; Bowman 2001].

User Action	Head Tracker Function	Mouse Function	Camera Control
Move/rotate head	Move/rotate (6DOF)	----	Move/rotate
Middle-click	----	Pick up/release object	Stays
Move/rotate hand	----	Move/rotate mouse (6 DOF)	Stays

Table 3.2 User Actions, Tracker Functions, and the Camera Control

3.2.4 Choice of Device for Navigation

In a head-coupled system, the user usually has an egocentric view and can easily investigate the environment by physical looking around. In spatial immersive display systems, users cannot see what is behind them and have to take advantage of some tools to do environment investigation. In addition, a user's movement has to be controlled by navigation tools.

Joysticks have been used in many studies for their simplicity and stability. In the previous study on comprising the steering modes [Chung 1992], subjects preferred to take a joystick as a natural navigation tool although the results revealed that it had no role in higher subject performance among operating a joystick, a mouse, and a spaceball™. In our work, a five-button digital joystick was chosen as the physical navigation device for its ease of use. It supports the egocentric wayfinding and egocentric travel that has the potential to minimize the user's load during trial. Hence, the subject might focus on the main process rather than on mastering complex control commands. The fly-through metaphor is registered to the joystick's movement (Table 3.3). The joystick controls the camera movement at three DOF, i.e., moving along the x- and y-axes, and rotation along the z-axis. Two speeds are allowed in our system.

Joystick Function	Camera Control
Move forward/backward	Forward/backward movement at lower speed
Move forward/backward + button 4	Forward/backward movement at higher speed
Left/right twist or left/right adjusting valuator	Turn left/right

Table 3.3 Joystick Functions and the Camera Control

3.3 Visual and Auditory Feedbacks

In the real world, the light, the sound emitted by different devices, signs, the sun, and streets are among the sensory cues accessible and used by people. As living creatures, we instinctively use these cues, interpreting them to create a mental picture or physical model of the world around us. In the virtual world, users rely on the virtual model, which defines all information that users can perceive, interpret, interact with, and most importantly, work in. Modeling the database can be difficult due to the process of classifying and evaluating visual cues. However, the designer should decide the relative importance of the visual cues in designing a virtual world [Bourdakis 1998]. We then can teach the subject to recognize and use these cues.

For providing effective visual feedback, we tried to build an information-rich model; that is, we tried to make the object "smart" [Kallmann & Thalmann 1999]. The virtual scenarios are drawn with color, shape, shadow, shading, and texture. These intrinsic physical properties can provide useful depth information and other visual stimuli. During exposure, the subject needs a way to know which object can be picked up. Therefore, we wire-framed the specific object to aid in the subject's recognition (Figure 3.7). Additionally, the specular highlights are rendered when the object is picked up; an image also appears on top of the destination platform when the matched object has been picked up (Figure 3.8). If the ball is dropped in range, the wire-frame, the highlight, and the image will disappear.

Visual feedback is enhanced by auditory feedback to increase realism. In our system, we have implemented sonification, using 2D sound to provide useful information. For example, different sounds are played when the visual database has been loaded or when the subject picks up an object, releases an object correctly or incorrectly, hits an obstacle, is sick, or finishes the trial. Thus, our system is a multi-sensory system since it integrates visual and auditory feedbacks. The subject is taught to understand the sound in the environment before trial. Table 3.4 lists all visual and auditory feedbacks supported by our system.

Visual Feedback	Auditory Feedback
Binocular disparity	Picks up an object
Perspective	Releases an object correctly / incorrectly
Occlusion	Hits an obstacle
Shadow, texture	Subject is uncomfortable
Motion parallax	End of the current session
	Loads the visual database

Table 3.4 Visual and Auditory Feedback

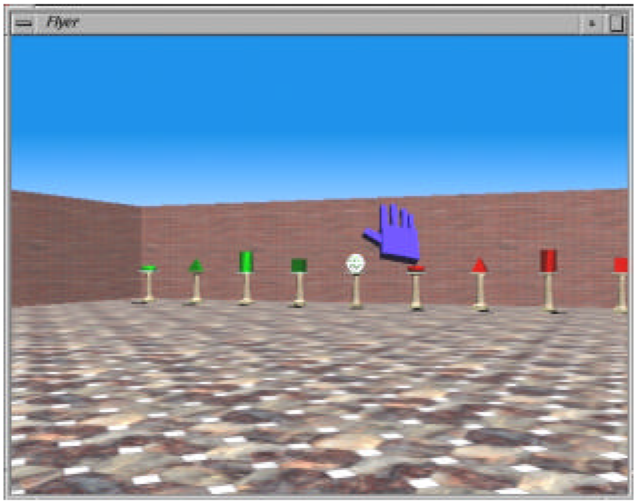


Figure 3.7 Visual Cue 1 – Wireframe
Objects have different shape, color, texture, shading, and shadow. They are important visual cues to increase scene realism. A green ball is wire-framed as a visual cue. To pick up the green ball, the virtual hand must be inside the visible wire frame.

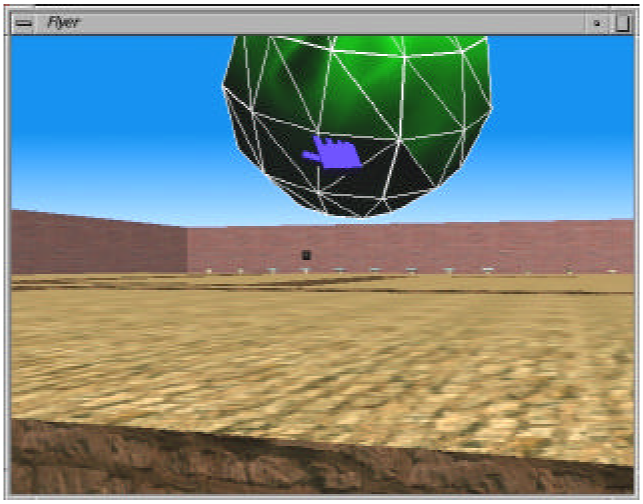


Figure 3.8 Visual Cue 2 – Highlight and Image
The green ball is highlighted if picked up. A path and a picture (upper the destination platform) appear immediately after pickup. The subject must walk through the path and release the green ball on the destination platform.

3.4 Pick-and-Release Tasks

The pick-and-release tasks are common physical actions performed in a virtual environment. One approach to understanding these tasks in VEs is to consider the frames-of-reference [Howard 1993] related to pick-and-release tasks. One frame is the *egocentric frame* that refers to some part of the observer, for example, head, hand or body of the user. The other is the *exocentric frame* that refers to the point external to the observer. An example of an exocentric judgment is to judge the absolute distance between two points in a room. A similar but egocentric judgment would be to report the distance between a point in a virtual room and the user's virtual body position.

Based on these categories, the pick-and-release tasks can be divided into: (1) body-centric navigation tasks, where the subject is to travel to a target in the virtual world by head tracking and joystick navigation; (2) hand-centric selection and hand-centric manipulation tasks, where the subject picks up an object, judges the distance, then releases the object on the destination platform. The subject operates a 3D mouse in performing the selection and manipulation tasks.

Two levels of scenario have been implemented for the user to perform pick-and-release tasks. Given an exposure session, two levels are loaded alternatively. Level one is relatively simple, since the obstacles are two obelisks sitting in the middle of the room besides the four room-walls and fifteen platforms. Additionally, the pickup task can be done without caring about the order of the objects. The subject may go to any platform, grab an object, then travel to the other side of the room, and release the object on the destination platform. In contrast, level two is more difficult. A maze-type wall is loaded as obstacles after an object is picked up. The subject has to avoid hitting the walls during navigation. Rather than to be picked up randomly, objects must be picked up in a predefined order. Given a level, the subject performs the pick-and-release tasks fifteen times to finish the current level. Then another level will be loaded. To help the subject perform pick-and-release tasks effectively, two metaphors are employed. One is for navigation, the other for selection and manipulation.

3.4.1 Navigation Metaphor

The *flying vehicle control metaphor* [Ware & Osborne 1990; Usoh & Steed 1995] is used in this work to aid in performing the body-centric navigation tasks. An advantage of using this metaphor is that physical locomotion is not required, so that the user can travel a long distance without leaving the seat. By manipulating the joystick, the subject can travel around the scenario. The head tracker used in the HMD system serves as a means to assist in wayfinding. In contrast, the subject is not head tracked in the Dome system. Thus, wayfinding has to be performed by operating a joystick. The mappings for both wayfinding and travel are linear.

Considering that it is important to implement constraints and limit DOFs without reducing the user's comfort significantly, we restricted travel to a fixed level relative to the floor of the room. Thus, the movement is along the x- and y-axes only. The head tracker, however, is operated in six DOF. Additionally, a mismatch of movement along the head direction might occur when the subject looks in other directions. Hence, we force the subject to look forward while traveling and to stop traveling while looking around. In certain situations, the subject can still fly around freely, look in any direction when staying still, and tilt his/her head in any orientation.

3.4.2 Selection and Manipulation Metaphor

To perform manipulation and selection, the classical *virtual hand metaphor* [Robinett & Holloway 1992] is employed in this work. We simply map the scale and position of the subject's physical hand directly to the scale and position of a virtual hand linearly for handcentric selection and manipulation. We do not use ray casting or the WIM technique because we do not want the subject's behavior in the virtual environment to go beyond the human's capability in the physical world. So at this point, the power of VEs is to duplicate the physical world, not to extend the subject's abilities to perform tasks impossible in the real world.

Chapter 4 Implementation

4.1 Libraries and Toolkits

Our system runs on an SGI Octane under the IRIX6.5.13 operating system [Http3]. The program is written in C/C++. The main simulation loop is built on top of the *OpenGL Performer™ library* [Rohlf & Helman 1994] version 2.2. It provides the features necessary for pursuing our design goals. *X/Motif* and *buttonfly* are the tools we used for developing user interfaces. Additionally, *gnuplot* is used to plot 2D graphs in our system.

Figure 4.1 illustrates all the possible dependencies between the numerous libraries, software tools, and applications. Dependency is indicated by the horizontal lines between components. A component above a line depends on the components below the line.

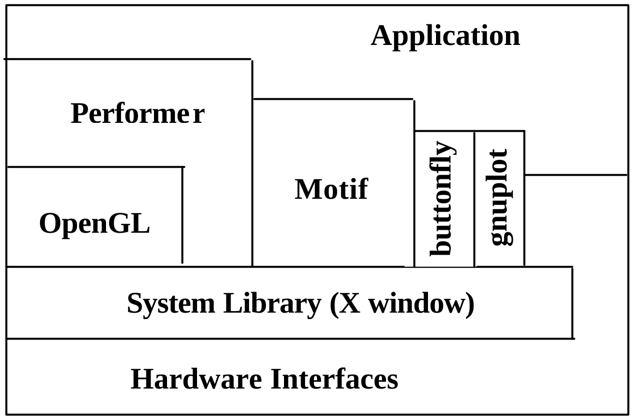


Figure 4.1 Software Libraries and Toolkits Dependency

4.1.1 OpenGL Performer Graphics Library

The OpenGL Performer™ 2.2 library supports multiple CPUs and provides a high-level graphics application programming interface (API). It also allows direct access to lower level rendering details and supports importing visual databases for more than thirty popular database formats.

The Performer core includes a visual database processing and rendering system. All data defined in the virtual world are held in a *scene graph*. The chain of the events in going from the scene graph to the display is illustrated in Figure 4.2. The processes are: (1) a visual database or scene is viewed by channels; (2) each channel's view of the scene is rendered by a pipe into a frame buffer; (3) each pipe defines a window that manages

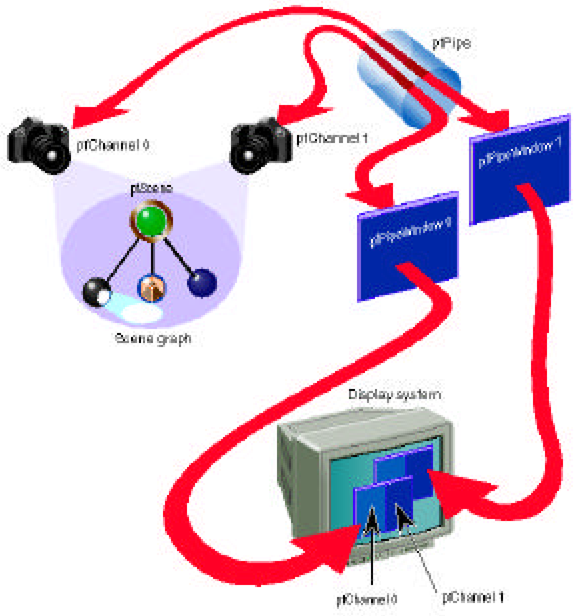


Figure 4.2 From Scene Graph to Visual Display

the frame buffer; (4) the images in the frame buffer are then transmitted to a display system that can be viewed by the subject [SGI].

4.1.2 Motif, Buttonfly, and Gnuplot

Motif is a library that can build all components of a GUI, for example, windows, buttons, and dialog boxes. Buttonfly is a user configurable, hierarchical, graphical menu system used to make pretty the user interface for Silicon Graphics [SGI] applications. Gnuplot is a command-driven interactive function plotting program. It can plot functions and data points in both 2D and 3D in many different formats. For example, we use gnuplot to output a series of 2D points to a graph in postscript format.

4.2 Simulator Architecture

The simulator architecture (Figure 4.3) is composed of seven modules: instructor/operator interface (IOI), the simulation loop, timing control, visual database management, input device management, data collection, and playback.

Instructor/operator represents the person who is in charge of the overall physical and virtual environments during exposure. His/her job is to take the subject through a carefully designed exposure procedure and record data. The *IOI,* a graphics user interface (GUI), provides a virtual platform for the operator. All commands can be issued through this GUI.

The *simulation loop* is the kernel of our system. It communicates with other five modules (except the playback module) when running the simulation. The loop repeats a series of actions for the duration of the main function. These actions manage the control of the application at each cycle. The simulation loop also captures events from a number

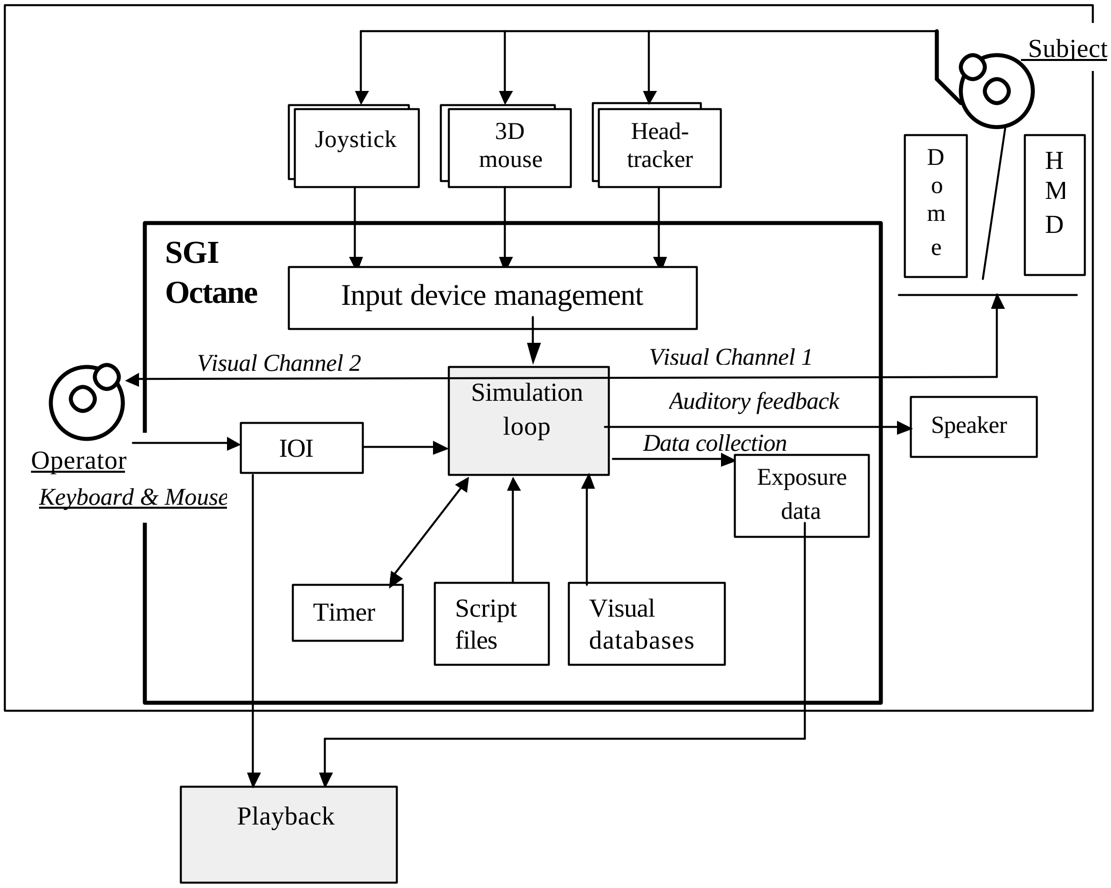


Figure 4.3 Simulator Architecture

of input devices, and then updates the visual and auditory feedbacks. The operator, who manages the running process, has the right to issue commands through the IOI interface to start or terminate the simulation loop.

During subject exposure, two levels are loaded alternatively. To decide which level is run, and how long the next simulation loop can be run, a *timer* module has been implemented. It controls the switch between level one and level two. Different visual databases and script files are loaded when running different levels. For example, assume the subject is exposed to a 30-minute session. When the program starts, the timer informs the simulation loop to load the database of level one and the corresponding script files. If the subject finishes this level in less than 12 minutes, the timer module will leave level one, load level two, and most importantly inform the simulation loop that 18 minutes remain. If the subject finishes level two in less than 18 minutes, the timer then restarts the level one loop. Otherwise, it will terminate this session.

Like other virtual environment applications, the system has integrated numerous input devices. The *input device management module* provides an interface to the simulation loop. At each cycle, the loop gets the input data from the input devices, e.g., joystick and the trackers. The sensorial output module captures these inputs and generates the corresponding visual and auditory stimuli. The subject experiences coherent feedback according to the instantaneous context. Finally, the data collection module records exposure data to the corresponding files.

The *playback module* is independent of the simulation loop. It gets the data from the data collection module and replays the exposure process in both 2D and 3D scenarios. The operator is capable of specifying an explicit subject id and session name through the IOI and replay the exposure process of that subject.

4.3 Visual Databases

4.3.1 Models

At level one (Figure 4.4), the model is a room about 21m long by 14m wide with a floor and 1.6m high four-side walls. On each side of the room, there are fifteen platforms. There are fifteen objects on one side of the platforms, in the shape of torus, pyramid, cylinder, box, and sphere, combined with the colors of green, red, and blue. The sizes of the objects are listed in Table 4.1. Two obelisks stand in the middle of the room as obstacles. Beside the basic geometry shapes, texture, shading, and shadow are created.

At level two (Figure 4.5), the room, objects, and platforms are the same size as in level one. Instead of using obelisks as obstacles, five different shapes of corridors are added (Figure 4.6). The paths of the corridors have almost the same number of left turns and right turns although the angles of the turns may be different. Again, texture, shading, and shadow are added.

Object	Size (X mm x Y mm x Z mm or R mm)
Room	21000x14000x4000
Torus	260x260x8
Pyramid	200x200x200
Cylinder	200x200x300
Box	180x180x180
Sphere	R = 240
Virtual hand	28x27x7
Level-one	
Column	350x350x1350
Level-two	
Paths	----

Table 4.1 Objects and Sizes

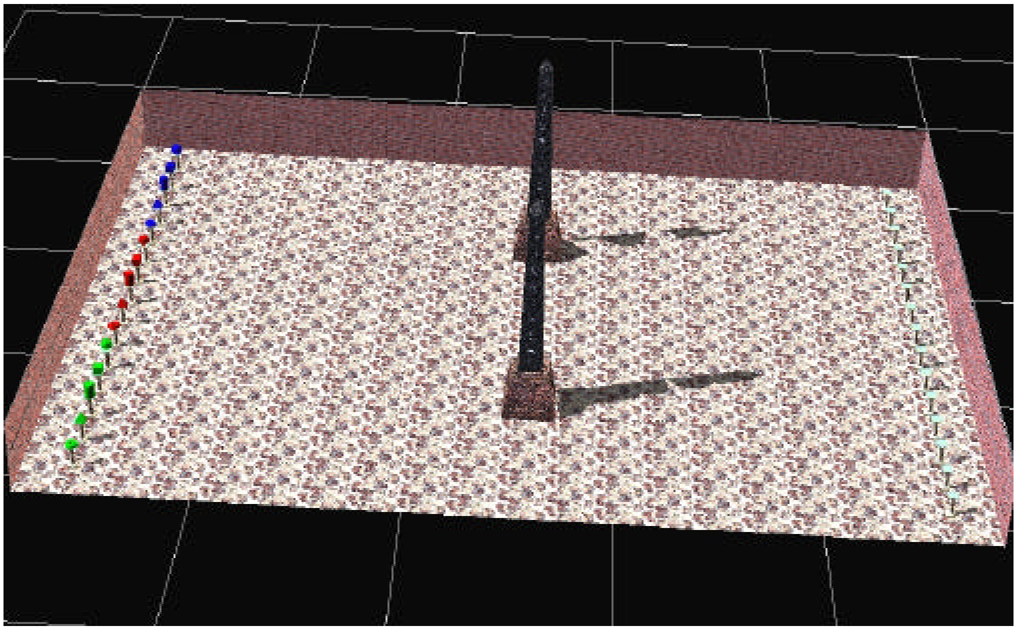


Figure 4.4 Level One Model
The scene includes 15 objects on one side of the room, 15 destination platforms on the other side, and two obelisks in the middle.

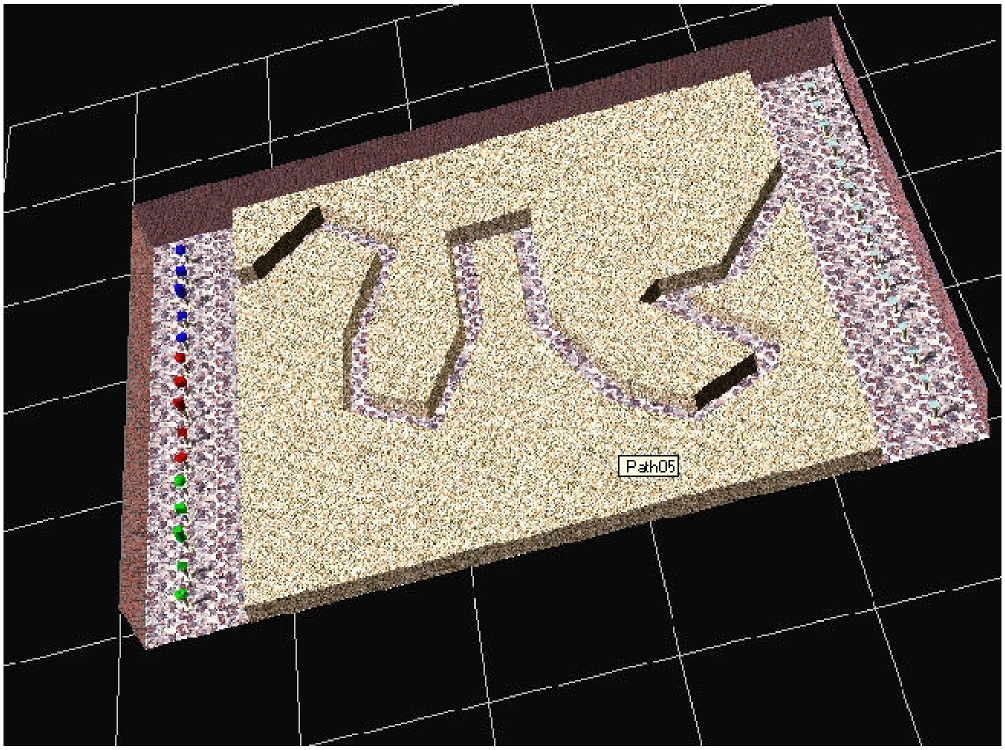


Figure 4.5 Level Two Model
It includes 15 objects on one side of the room, 15 destination platforms on the other side. Path No. 5 is shown here.

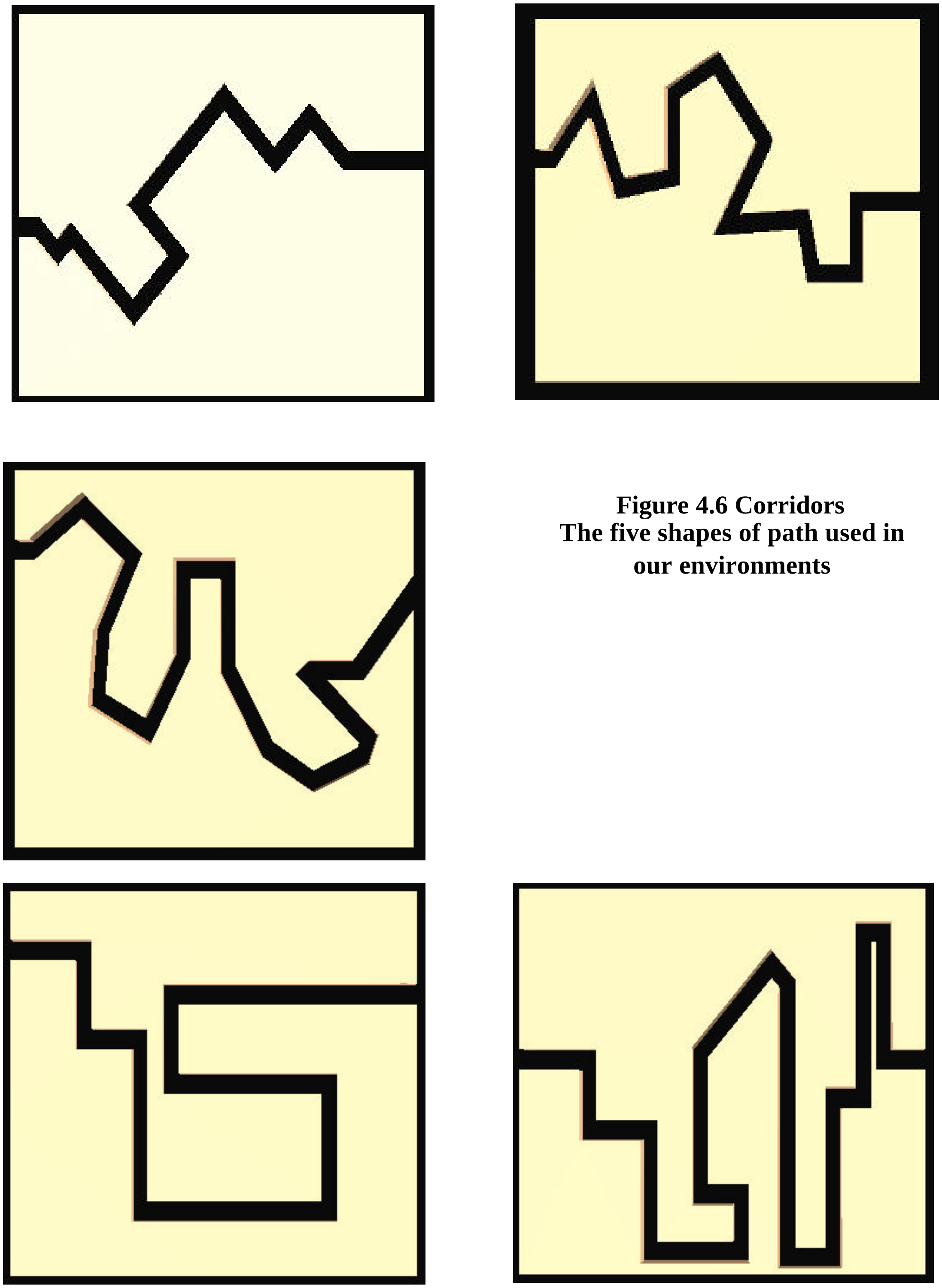


Figure 4.6 Corridors
The five shapes of path used in our environments

4.3.2 ASE Scene Generator

The ASE (ASCII Scene Export) Scene Generator was written to render the database specified by the ASE file into a Performer scene. Each object in the ASE file is a node that has its world coordinates. The reason we use this generator to create the scene graph rather than importing the 3D Max models directly is because we want to control the objects' behaviors separately without loading many 3D objects in separate data files. In addition, the rendering was implemented in this generator. The drawback of the ASE parse, however, is that it does not handle backface culling. All polygons and textures belonging to an object are rendered to the scene graph no matter whether they can be seen or not from the current viewpoint. This puts additional burden on the rendering process.

4.4 Simulation Loop

The simulation loop (Figure 4.7) is a Performer program. It first reads a set of script files including an *application script*, a *player script*, and a *level script*. In the script files, we can specify the files to be loaded, subject information (e.g., hand-head distance, arm distance), scene database, and the objects' behaviors (e.g., movable object, highlight). Each script file is given numerous tokens to define different behaviors. The main Performer function parses these scripts and renders the scene. Once a subject interacts with the scene, it will be updated.

```
#include headerfiles

void main()
{
   pfInit( );
   pfConfig( );
   InitDevices( );
   pfScene ( );  //initiate the scene
   loadLevel ( ); // defines current level

   loadPlayer( ); // defines a player (subject)

   Timer_begin( ); // begin timer
   while (not finish)
   {
        read joystick;     // read input device
        read tracker;
        read mouse;
        read keyboard;   // determine interaction

        update subject state;
        update scene;      // update object state, and sensorial management
        write to output;   // general output management

        pfSync( );
        pfFrame( );
   }
   Timer_end( );
```

Figure 4.7   Pseudo Code of the Simulation Loop

We have implemented two views, one for the subject and the other for the operator. Using Performer™ based functions, we create two independent pipelines in order to configure the two views; they access the same-shared scene definition. The code segment is shown in Figure 4.8.

```
pfInit( );
……
pfMultipipe(2);           // initialize to support two pipes
……
pfConfig( );

pfPipe *p[2];
pfChannel *chanPtr[2];

for (int ii=0; ii<2; ii++)
{
   p[ii] = pfGetPipe(ii);
   p[ii]->setScreen(ii);    // one screen to the subject, the other to the operator
   initPipe(p[ii]);          // setup pipe parameters
   chanPtr[ii] = new pfChannel(ii);
}
 chanPtr[0] ->attach(chanPtr[1]);
```

Figure 4.8 Multiple View Implementation

We also created some visual effects. An earth-sky model was defined and attached to the display channels, so that Performer™ can automatically render the scene with blue sky, purple horizon, and yellow ground. Other visual effects, for example, highlighted and wire-framed bounding boxes, were also implemented by calling Performer™ functions. Textures and shadows were static and were created at the model creation stage because Performer™ 2.2 does not support dynamic ones. The newest version, Performer™ version 2.4, can load shadows and textures dynamically at run time. This new library provides a new class called “Shader” which supports appealing visual attributes. In addition, collision detection is performed by testing an axis-aligned bounding box around the subject’s virtual body against all virtual obstacles in the environment. The auditory feedback is rendered by a system call to play sound.

4.5 Distortion Correction

To render correct images on the spherical surface of dome, we must implement distortion correction. The basic idea of distortion correction is to utilize texture-mapping techniques. The data are first rendered to the six faces of a cube from the scene centroid (Figure 4.9). The six images are then used as textures and are warped to create a sphere projection (Figure 4.10). Finally, the projected sphere images are rendered to the eye space, that is at which the user is looking.

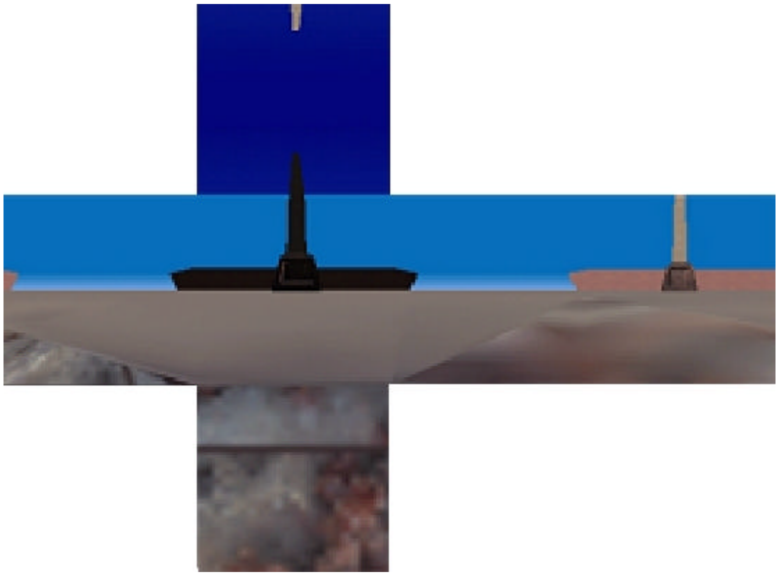


Figure 4.9 Distortion Correction – Create Six Faces of a Cube.
From left to right are left, front, right, and back faces. From top to bottom are top, front, and bottom faces.

The Elumens Spherical Projection of Image (SPI) API uses a similar approach, called *Tri-map*. It only renders the front, the right half of the left and the left half of the right faces since the displayed horizontal FOV is 180^0 rather than 360^0. This accelerates the rendering speed significantly because the data are sent down the graphics pipeline only three times. Full circle projection uses the frame buffer about 59%. To use the buffer more efficiently, the Tri-map approach produces images in the shape of a truncated circle (Figure 4.11). The image is enlarged and uses 79% of the frame buffer.

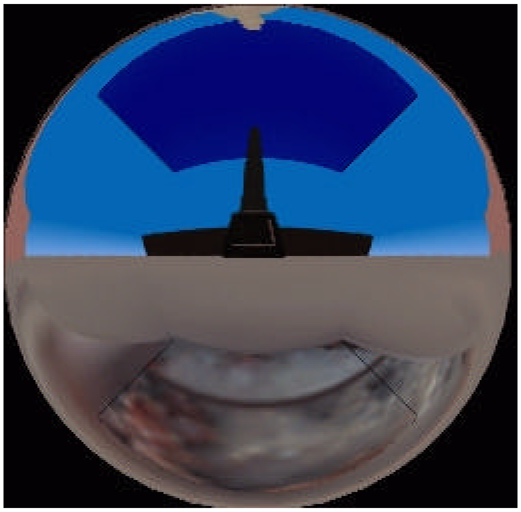


Figure 4.10 Sphere Mapping

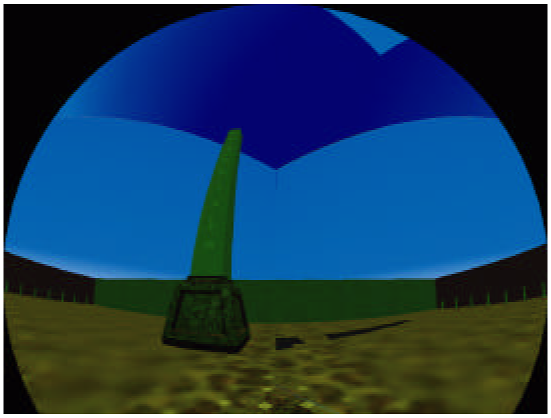
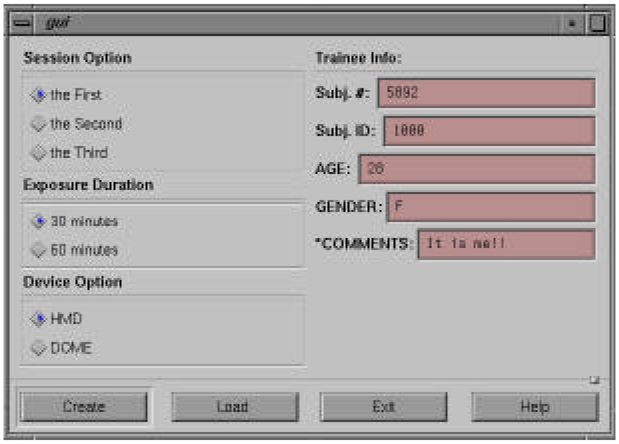


Figure 4.11 Truncated Sphere Mapping

4.6 Instructor / Operator Interfaces

Clearly, many operators are not programmers. As a result, an Instructor/Operator Interface (IOI) was built. It is composed of a series of menus and buttons and serves as a virtual platform between the operator and the simulator. Most importantly, the IOI gives a hint of the order of the operations the operator must carry out. For example, hardware tests must be done before starting the subject test, so that we put the *hardware test menu* before the *trial menu*. The operator only needs to go through these steps to finish the trial without typing any commands from the shell window.

The IOI first brings the operator to execute the hardware test. A *listening to the sound menu* is used to train the subjects to understand the auditory cues they may receive. Then a *subject information menu* (Figure 4.12) needs to be filled in with the subject ID, the type of the display device, the exposure information, etc. To select a subject identification to be exposed or delete a subject, the operator can open a *subject list menu* and a *remove subject menu*. If selecting an identification to be exposed, the following exposure data will be saved in a data file. The data can then be loaded from the *playback menu* used by



Session Option
● the First
○ the Second
○ the Third

Exposure Duration
● 30 minutes
○ 60 minutes

Device Option
● HMD
○ DOME

Trainee Info:
Subj. #: 5892
Subj. ID: 1000
AGE: 26
GENDER: F
*COMMENTS: It is me!!

Figure 4.12 Subject Information Menu

the operator to review this subject's exposure process. A *reset* button is created as well which can be used by the operator to terminate the current trial.

4.7 Data Collection and Playback Design

Our system provides the mechanism for saving and retrieving trials. It allows the operator to go back, reexamine the trial process, and show them to the subject or collaborators. It is also an important part of using visualization techniques to get a better understanding of the subject's performance.

4.7.1 Data Collection

Traditional data collection is done using videotaping or a written account. Subjective reporting using questionnaires is a very common technique [Kennedy et al. 1993]. It has been proved useful in the postural stability research [Kennedy & Lilienthal 1995]. However, it is done *after* the trial and we must assume that the subject retains detailed memories of each part of the experience. Unfortunately, this is not always true. Our method complements this technique by recording data *while* the subject is immersed in the virtual environment. They can be used together with videotape and written documents to investigate subject performance. The data collection module is implemented in software without interfering with the subject exposure. The data we record include various aspects of the trial. This includes task completion time, task accuracy, current subject's position and orientation in each loop, the name of the current picked object, the name of the dropped platform, collision errors, completion time, selection errors, and sickness status.

4.7.2 Playback Design

Once the exposure data are sampled, we need a way to simulate the exposure process. The playback module implements this function. We present two views: one has been implemented as a 2D graph, and the other is a 3D view. The 2D graph is drawn by *gnuplot* and displays the path the subject flew through during a selected exposure (Figure 4.13). The 3D view presents both overviews (global view) and a life-size virtual environment (local view) (Figure 4.14). This replay duplicates the subject's operations during the trial. For example, if the subject picked up an object, the object also moved the same way the subject picked up the object. When the subject hits an obstacle or a wall, the corresponding sound is played to indicate the collision. The Motif interface in the figure is run by a forked process. After loading the scene, one process handles rendering the scene, and another process is forked for checking the interface status.

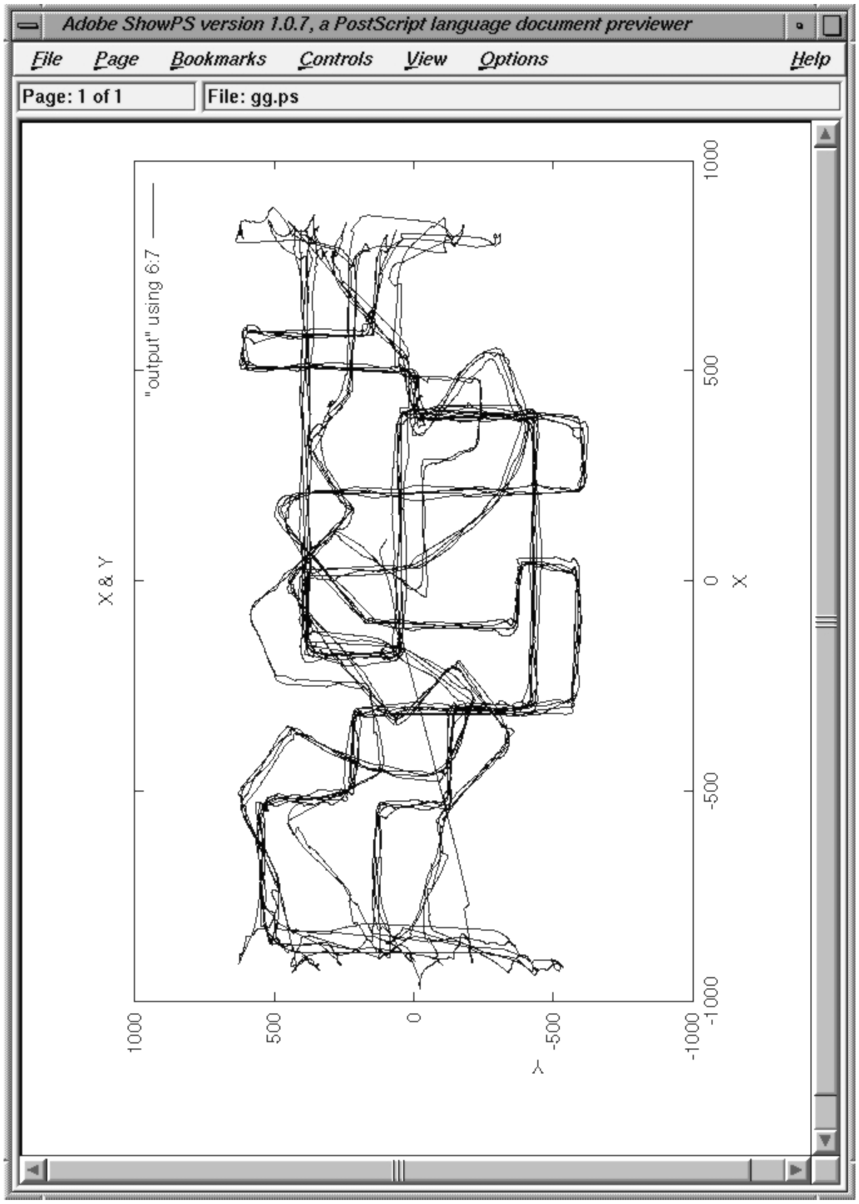
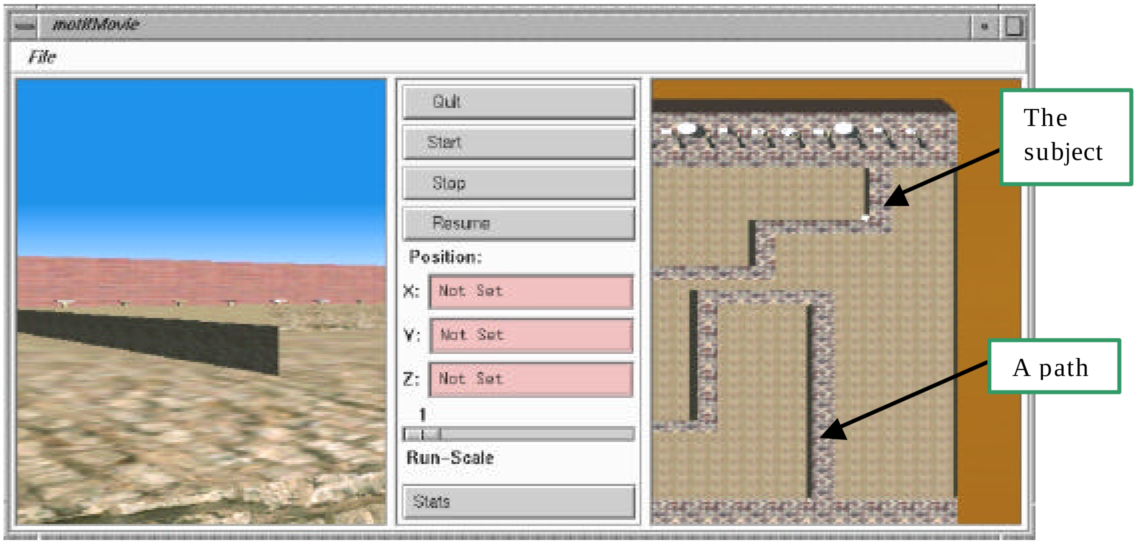


Figure 4.13 2D Plot of Exposure Data (Level Two)
The 2D plot draws the paths through which the subject has flown during the exposure



X: Not Set
Y: Not Set
Z: Not Set

Figure 4.14 3D Plot of Exposure Data (Level Two)
The left picture shows the local view of the subject and the right picture gives a global view. The subject is shaped with the shape of the picked object. The operator can left click mouse button to change the viewpoint along the current plane. The middle part is a motif GUI.

Chapter 5 Results and Future Work

5.1 Results

We present our results by giving two examples of a subject exposure. In the level one example, the subject picks up a red ball and releases it on its destination platform; in the level two example, the subject picks up a green ball and releases it on its destination platform.

5.1.1 Level One

The subject's task is to walk to the left side of the environment, pick up an object, then walk to the right side, and release the object. The subject is asked to perform the task at a quick and comfortable pace while minimizing errors. During travel, the subject should try not to hit the walls and other obstacles in the virtual world. A trial is terminated when the subject runs out of time or feels sick.

All objects are initialized outside the field of view. Figure 5.1 shows the relative

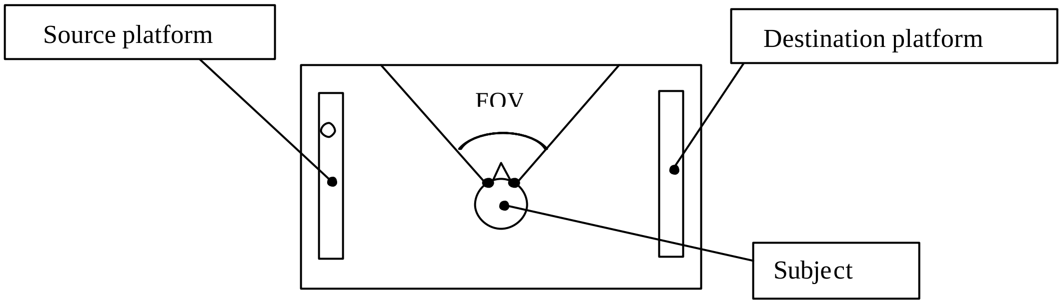
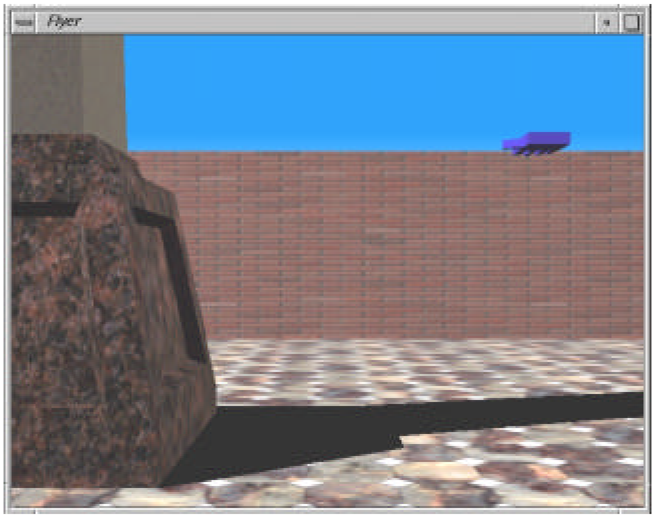


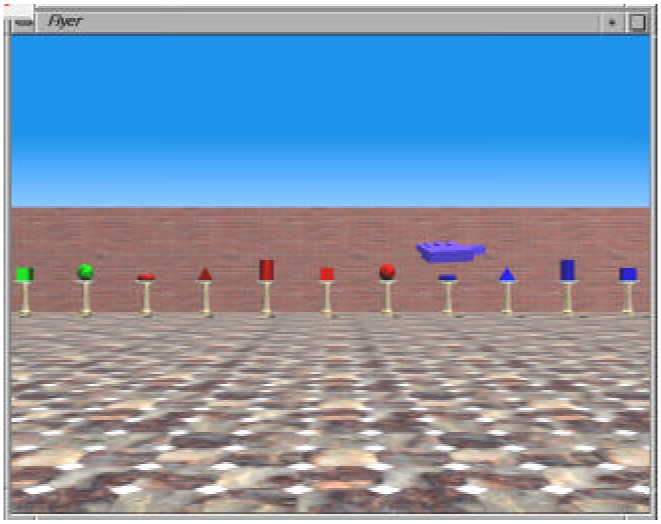
Figure 5.1 Initial View of the Subject

position and orientation of the subject in the virtual world. The subject is situated in the middle of the room and faces a wall (Figure 5.2a). The subject has to turn his/her head to locate the target objects visually (Figure 5.2b). The body-centric navigation task measures a subject's ability to travel through an environment while avoiding obstacles. In this case, the travel is performed by joystick manipulation and the wayfinding by head tracking. Two constant travel speeds are allowed to avoid potential simulator sickness. Three dimensions of travel freedoms – moving along the x- and the y- axes, and rotating along the z-axis, are performed by joystick manipulation. In the HMD system, the head tracker supports the six degrees-of-freedom wayfinding task. When the virtual hand touches an object on a platform (Figure 5.2c), the object can be picked up and attached to the virtual hand (Figure 5.2d). The destination platform is wire-framed as the visual cue (Figure 5.2e).



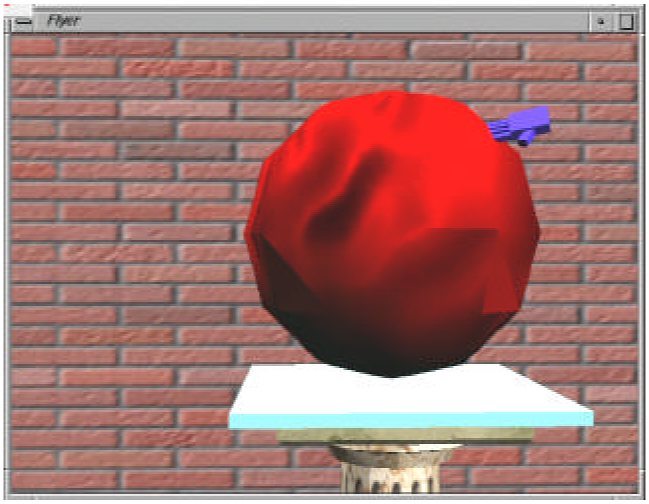
a

No target object can be seen at first glance.



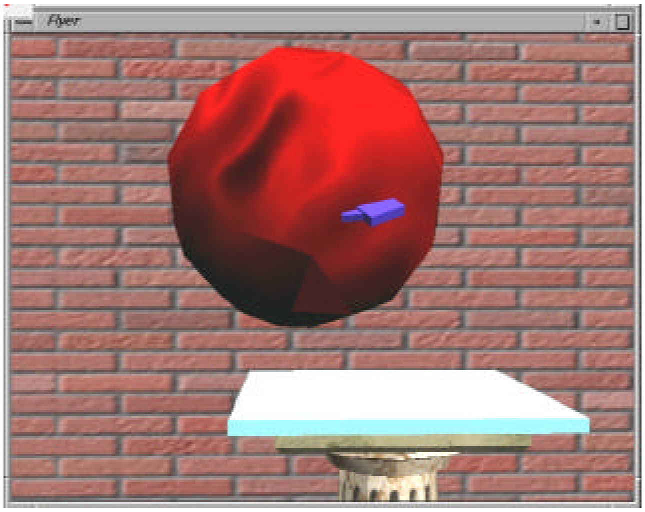
b

The subject rotates his/her head physically or manipulates the joystick to find the object that can be picked up.



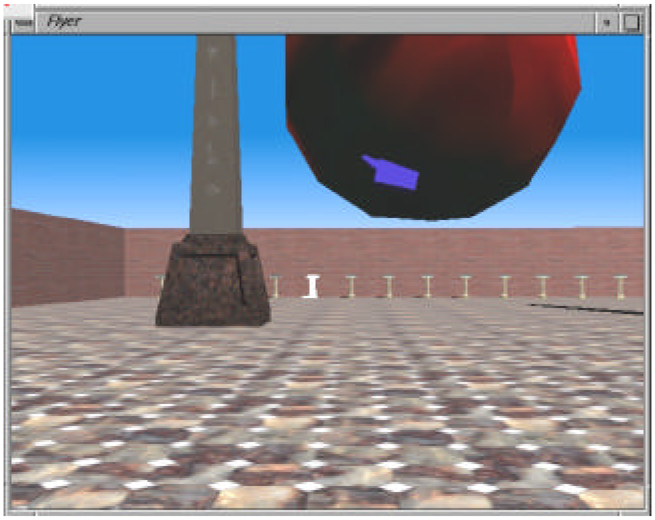
c

By manipulating the joystick, the subject flies through the scene and moves to a red ball. The virtual hand then touches the ball.



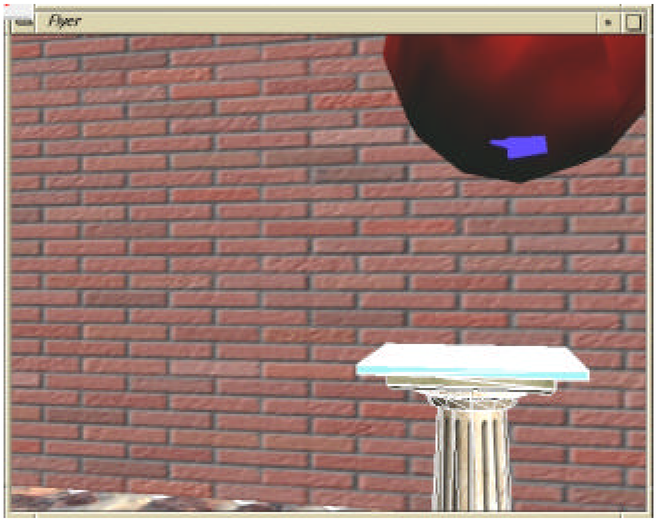
d

The subject middle-clicks and picks up the ball. The ball is attached to the virtual hand. A sound is played simultaneously.



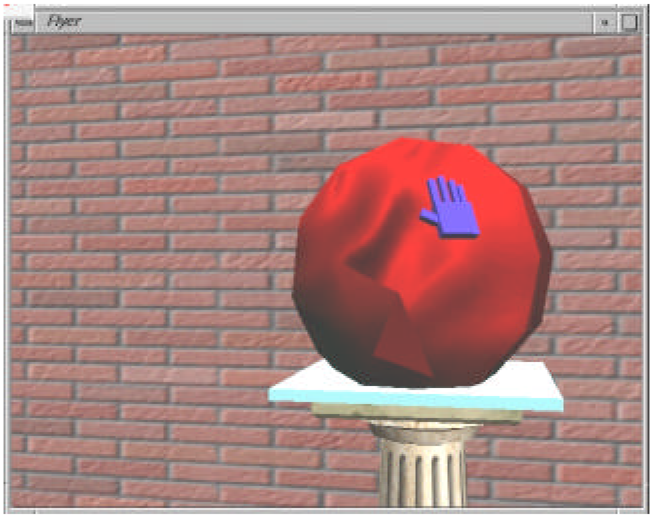
e

The subject is walking towards the destination platform that has been highlighted.



f

The subject is approaching the destination platform.



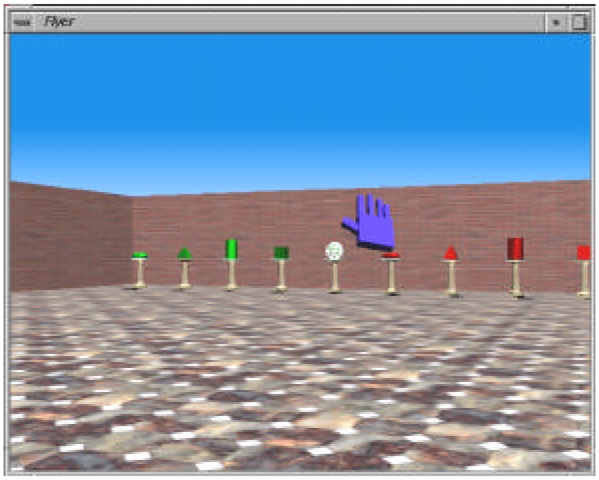
g

The subject has released the object on the matched platform. The highlight disappears. A sound is played simultaneously.

Figure 5.2 Example of Pick-and-Release of a Red Ball (Level One)

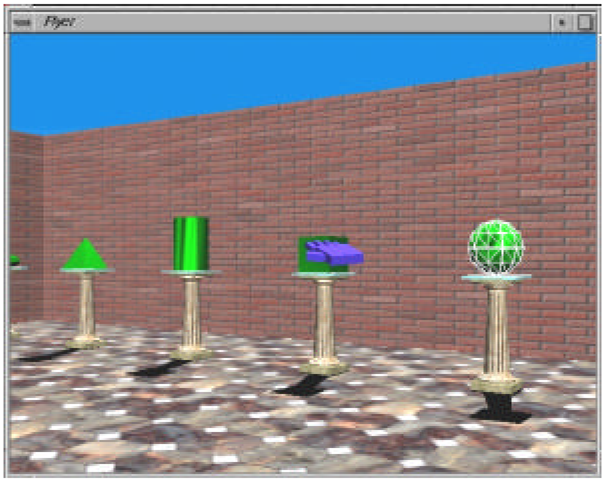
5.1.2 Level Two

In level two, the corridors serve as obstacles. The functions of the joystick, the head tracker, and the corresponding camera movement are the same as those in level one. The objects need to be picked up in a predefined order.



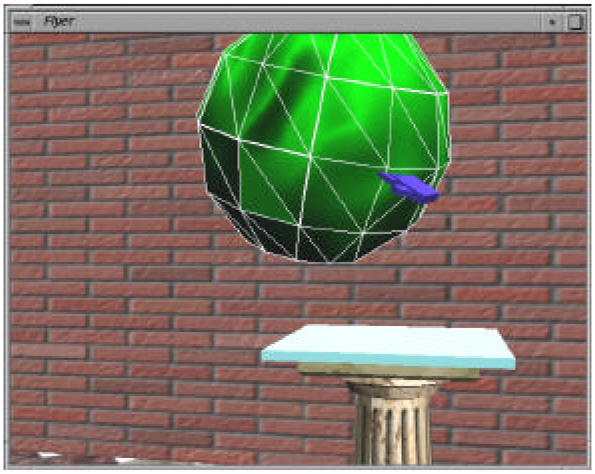
a

The green ball is wire-framed. This is the first object the subject should pick up.



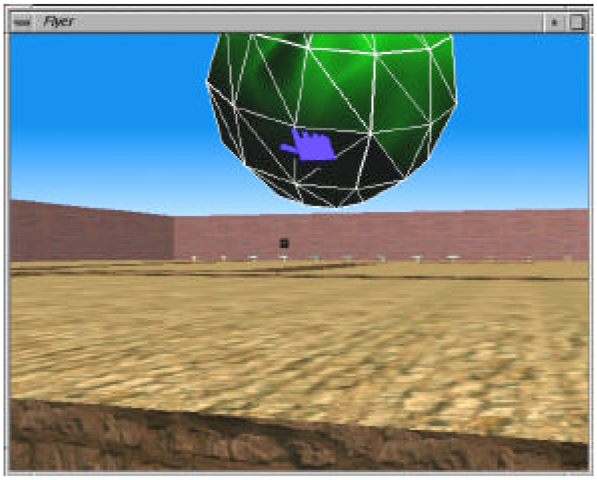
b

The subject is approaching the object by joystick operation.



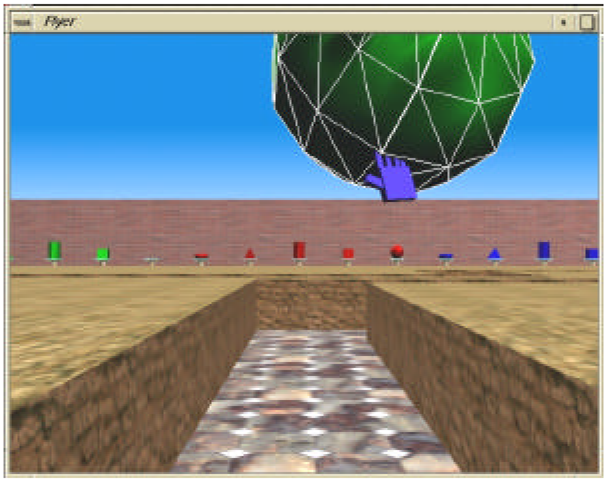
c

The virtual hand touches the object. The subject middle-clicks and picks up the green ball.



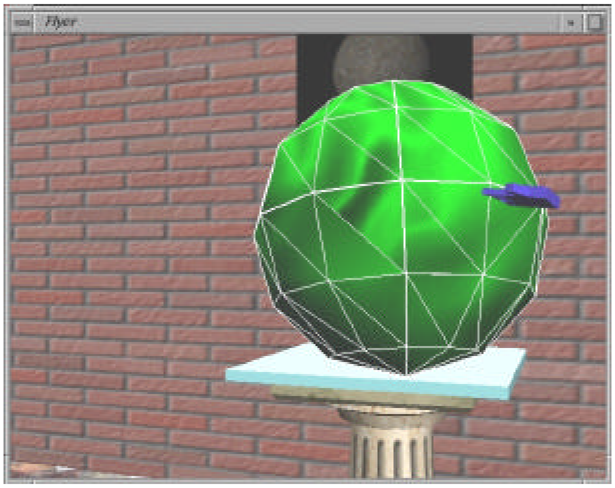
d

A path and an image appear. The subject needs to put the ball on the destination platform on which there is an image.



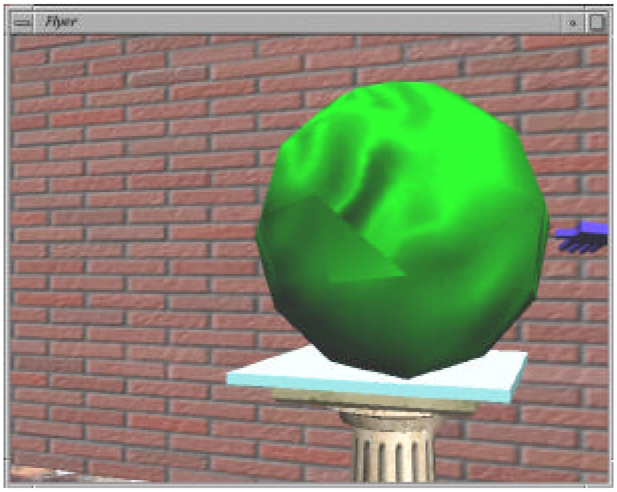
e

The subject is flying through the path. Collision errors are recorded.



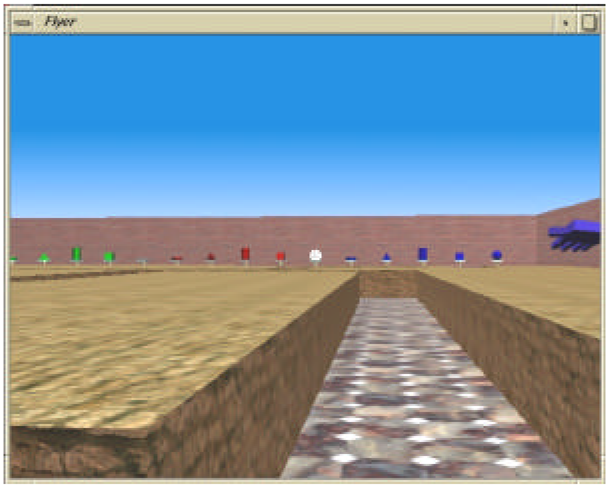
f

The subject manipulates the 3D mouse and adjusts the position of the green ball. The nearer the dropped position is to the center of the platform, the better.



g

The subject middle-clicks to drop the ball. The image and the bounding box of the ball disappear. Sound is played.



h

The subject should follow the path back and pick up the next wire-framed object.

Figure 5.3 Example of Pick-and-Release of a Green Ball (Level Two)

5.2 Conclusion

Motivated by the need for a comparative study of the VR4 HMD and the projector dome regarding pick-and-release tasks, we present a virtual environment system to help the evaluation from the point of view of both displays and the interaction tasks. The system is flexible, solid, and easy to use. It serves as a testbed and can be further used by other studies.

Creating an effective and efficient virtual environment is a very complex process, from modeling to interaction design. The techniques proposed here to attain an effective environment include:

- A formalized design method
- Modeling the scene using *smart* object technique
- Efficient data collection and playback features

None of these ideas is particularly new. Rather, the main contribution of the present work is that it combines the solutions into a complete system. Except for the study by Bowman and coworkers [2001], we are not aware of any formal studies that compare state-of-art display devices. We have taken advantage of previous interaction research and demonstrated how to build the display comparison systems, derived from task analysis, choices of various devices, and interaction tasks.

5.3 Future Work

This section points out several directions for future work. One of the dreams driving this work is to extend current systems to a uniformed virtual environment testbed. At this point, it should support all known interaction components, and integrate various input and output devices. Besides making additions to the application, an important area of future work is to conduct a formal user evaluation. One goal of the formal evaluation is to determine the benefits of display devices; another goal is to validate the appropriateness of the interface design.

One of the future works is about the level-of-detail and hidden-surface removal. We did not consider this in the *ASE scene generator*. However, it puts a burden on the hardware and reduces frame rates. The other thinking is to do the comparison in another way: rather than considering building a system with minimal difference, we can build a system with the best suitable interface. For example, the dome system is much more likely to provide an exocentric view. Egocentric wayfinding is harder. We can provide a WIM to assist in the user's wayfinding. The small world does not necessarily need to appear in the HMD system. The small world is not necessary to be locked on the screen all the time since it occludes part of the user's view.

The perspective view of the destination platform makes the precise release hard. The six degree-of-freedom hand tracking restricts the user to the coarse release of virtual objects. However, precise release might be acquired by providing a top view of the destination platform without losing the DOF of the 3D mouse. We can implement it by checking the object position relative to the destination platform. If the distance is in a pre-defined range, a map pops up automatically with the top view of the platform and the object in real-time. We can still use the previous map by changing the image since we don't need to care about our position in the virtual world when dropping the object.

References

Acosta, E., Stephens, B., Temkin, B., Krummel, T.M., Gorman, P.J., Griswold, J.A., and Deeb, S.A. (1999), "Development of a Haptic Virtual Environment," *Proc. of the IEEE Symposium on Computer-Based Medical Systems*, 35-39.

Applewhite, H. (1991), "Position Tracking in Virtual Reality," *Proc. of the Virtual Reality, Beyond the Vision: The Technology, Research, and Business of Virtual Reality*, 1-8.

Arsenault, L.E. and Kelso, J. (2002), "The DIVERSE Toolkit: A Toolkit for Distributed Simulations and Peripheral Device Services," *To appear at Proc. of IEEE VR'2002*.

Azuma, R. and Bishop, G. (1994), "Improving Static and Dynamic Registration in a See-Through HMD," *SIGGRAPH 1994 Conference Proceedings*, 197-204.

Azuma, R. and Bishop, G. (1995), "A Frequency-Domain Analysis of Head-Motion Prediction," *SIGGRAPH 1995 Conference Proceedings*, 401-408.

Barfield, W. and Weghorst, S. (1993), "The Sense of Presence within Virtual Environments: A Conceptual Framework," *In G. Salvendy and M. Smith (Eds.), Human-Computer Interaction: Software and Hardware Interfaces*, Amsterdam: Elsevier, 699-704.

Barfield, W., Hendrix, C., and Bystrom, K. (1997), "Visualizing the Structure of Virtual Objects Using Head Tracked Stereoscopic Displays," *Proc. of the Virtual Reality Annual International Symposium (VRAIS'97)*, 114-119.

Begault, D. (1994), "3D Sound for Virtual Reality and Multimedia," *Academic Press*.

Bishop, G., Bricken, W., Brooks, F., Brown, M., Burbeck, C., Durlach, N., Ellis, S., Fuchs, H., Green, M., Lackner, J., McNeill, M., Moshell, M., Pausch, R., Robinett, W., Srinivasan, M., Sutherland, I., Urban, D., and Wenzel, E. (1992), "Research Directions in Virtual Environments," *Report of an NFS Invitational Workshop*, UNC, 154-189.

Bourdakis, V. (1998), "Navigation in Large VR Urban Models," *Proc. of the Virtual Worlds 98*, 345-356.

Bowman, D., Koller, D., and Hodges, L.F. (1997), "Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques," *Proc. of the Virtual Reality Annual International Symposium (VRAIS'97)*, 45-52.

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

Bowman, D., Wineman, J., Hodges, L.F., and Allison, D. (1998), "Designing Animal Habitats Within an Immersive VE," *IEEE Computer Graphics and Applications*, 18(5): 9-13.

Bowman, D. and Hodges, L.F. (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, D., Johnson, D., and Hodges, L.F. (2001), "Testbed Evaluation of Virtual Environment Interaction Techniques," *Presence*, 10(1): 75-95.

Bowman, D., Kruijff, E., LaViola, J.J., and Poupyrev, I. (2001), "An Introduction to 3-D User Interface Design," *Presence*, 10(1): 96-108.

Bowman, D. and Wingrave, C.A. (2001), "Design and Evaluation of Menu Systems for Immersive Virtual Environments," *Proc. of the IEEE Virtual Reality*, 149-156.

Bowman, D. (2001), "Advanced Topics in 3D User Interface Design: Basic 3D interaction Techniques," *ACM SIGGRAPH Course Notes No. 44*.

Bowman, D., Datey, A., Farooq U., Ryu, Y.S., and Vasnaik, O. (2001), "Empirical Comparisons of Virtual Environment Displays," *Tech. Report TR01-19*, Virginia Polytechnic Institute & State University, Dept. of Computer Science, VA.

Brelsford, J. (1993), "Physics Education in a Virtual Environment," *Proc. of the Human Factors and Ergonomics Society 37th Annual Meeting*, 1286-1290.

Brooks, F.P. (1986), "Walkthrough – A Dynamic Graphics System for Simulating Virtual Buildings," *Proc. of the ACM Workshop on Interactive 3D Graphics*, 9-21.

Brooks, F.P. (1988), "Grasping Reality through Illusion – Interactive Graphics Serving Science," *Proc. of the ACM CHI'88*, 1-11.

Brooks, F.P., Ouh-Young, M., Batter, J.J., and Kilpatrick, P.J. (1990), "Project GROPE: Haptic Displays for Scientific Visualization," *Computer Graphics*, 24(4): 177-185.

Brooks, F.P. (1999), "What's Real about Virtual Reality," *IEEE Computer Graphics and Applications*, 19(6): 16-27.

Bryson, S. and Levit, C. (1991), "The Virtual Wind Tunnel: An Environment for the Exploration of Three-Dimensional Unsteady Flows," *Proc. of the Visualization'91*, 17-24.

Chen, J., Fang, Y., Loftin, B.L., Leiss, E.L., Lin, C., and Su, S. (2001), "An Immersive Virtual Environment Training System on Real-Time Motion Platform," *Proc. of the Conference on Computer Aided Design and Computer Graphics*, China, 951-954.

Chung, J.C. (1992), "A Comparison of Head-tracked and Non-Head-Tracked Steering Modes in the Targeting of Radiotherapy Treatment Beams," *Proc. of the ACM Symposium on Interactive 3D Graphics*, 193-196.

Cobb, S.V., D'Cruz, M.D., and Wilson, J.R. (1995), "Integrated Manufacture: A Role for Virtual Reality", *International Journal of Ergonomics*, 16:411-425.

Conner, D., Snibbe, S., Herndon, K., Robbins, D., Zeleznik, R., and van Dam, A. (1992), "Three-Dimensional Widgets," *Proc. of the 1992 Symposium on Interactive 3D Graphics*, 197-208.

Cruz-Neira, C., Sandin, D.J., Defanti, T.A., Kenyon, R.V., and Hart, J.C. (1992), "The Cave Audio Visual Experience Automotive Virtual Environment," *Communications of the ACM*, 35(6): 64-72.

Cruz-Neira, C., Sandin, D.J., and Defanti, T.A. (1993), "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," *SIGGRAPH 1993 Conference Proceedings*, 135-142.

Cruz-Neira, C. and Lutz, R.R. (1999), "Using Immersive Virtual Environments for Certification," *IEEE Software*, 16(4): 26-30.

Darken, R.P. and Sibert, J.L. (1993), "A Toolset for Navigation in Virtual Environments," *Proc. of the ACM Symposium on User Interface Software and Technology (UIST'93)*, 157-165.

Darken, R.P., Cockayne, W.R., and Carmein, D. (1997), "The Omni-directional Treadmill: A Locomotion Device for Virtual Worlds," *Proc. of UIST'97*, 213-221.

Darken, R.P., Hix, D., Durlacj, N., Dittmar, M.L., and Bolas, M. (1998), "Usability is from Venus, Virtual Environments are from Mars: User-Centered Evaluation Methodology in a New Medium," *Proc. of the Virtual Reality Annual International Symposium (VRAIS'98)*, 218.

Davis, E.T. and Hodges, L.F. (1995), "Human Stereopsis, Fusion, and Stereoscopic Virtual Environments," *Virtual Environments and Advanced Interface Design*, Oxford University Press, 145-174.

Dean, K.L., Asay-Davis, X.S., Finn, E.M., Foley, T., Friesner, J.A., Imai, Y., Naylor, B.J., Wustner, S.R., Fisher, S.S., and Wilson, K.R. (2000), "Virtual Explorer: Interactive Virtual Environment for Education," *Presence*, 9(6): 505-523.

Dinh, H., Walker, N., Hodges, L.F., Sone, C., and Kobayashi, A. (1999), "Evaluating the Importance of Multi-Sensory Input on Memory and the Sense of Presence in Virtual Environments," *Proc. of the IEEE Virtual Reality'99*, 222-228.

Dede, C., Salzman, M., and Loftin, R.B. (1996), "ScienceSpace: Virtual Realities for Learning Complex and Abstract Scientific Concept," *Proc. of the Virtual Reality Annual International Symposium (VRAIS'96)*, 246-252.

Durlach, N.I. (1991), "Auditory Localization in Teleoperator and Virtual Environment Systems: Ideas, Issues, and Problems," *Perception*, 20: 543-554.

Durlach, N.I., Allen, G.L., Darken, R.P., Garnett, R.L., Loomis, J., Templeman, J., and von Wiegand, T.E. (2000), "Virtual Environments and the Enhancement of

Spatial Behavior: Towards a Comprehensive Research Agenda," *Presence*, 9(6): 593-615.

Edgar, G.K. and Bex, P.J. (1995), "Vision and Displays", In K. Carr and R. England (Eds.), Simulated and Virtual Realities - Elements of Perception, London: Taylor and Francis.

Fechter, J., Grunert, T., Encarnação, L.M., and Strasser, W. (1996), "User-Centered Development of Medical Visualization Applications: Flexible Interaction through Communicating Application Objects," *Computers and Graphics*, 20(6): 763-774.

Fisher, S. (1989), "The AMES Virtual Environment Workstation (VIEW)," *SIGGRAPH 1989 Course Notes No. 29*.

Foley, J., Wallace, V., and Chan, V. (1984), "The Human Factors of Computer Graphics Interaction Techniques," *IEEE Computer Graphics and Application*, 4(11): 13-48.

Foley, J., van Dam, A., Feiner, S., and Hughes, J. (1990), "*Interactive Computer Graphics: Principles and Practice,*" MA: Addison-Wesley.

Forsberg, A., Herndon, K., and Zeleznik, R. (1996), "Aperture Based Selection for Immersive Virtual Environment," *Proc. of the ACM Symposium on User Interfaces Software and Technology*, 95-96.

Gabbard, J.L., Hix, D., and Swan II, J.E. (1999), "User-Centered Design and Evaluation of Virtual Environment," *IEEE Computer Graphics and Application*, 19(6): 51-59.

Gomez, D.G. and Langrana, N. (1995), "Integration of the Rutgers Master II in a Virtual Reality Simulation," *Proc. of the Virtual Reality Annual International Symposium (VRAIS'95)*, 198-202.

Harm, D.L. (1998), "Virtual Reality Training: Cybersickness and Effects on Sensorimotor Functions," *Tech. Report NRA98-HED-02, NASA*.

Henry, D. and Furness, T. (1993), "Spatial Perception in Virtual Environments: Evaluating an Architectural Application," *Proc. of the Virtual Reality Annual International Symposium (VRAIS'93)*, 33-40.

Hinckley, K., Pausch, R., Goble, J., Kassell, N. (1994), "Passive Real-World Interface Props for Neurosurgical Visualization," *Proc. of CHI'94*, 452-458.

Hirose, M. (1995), "The Synthesis of Realistic Sensations," *Proc. of the Fifteenth International Display Research Conference*, Japan, 17-20.

Hix, D. and Hartson, H.R. (1993), "Developing User Interfaces: Ensuring Usability Through Product and Process," NY: John Wiley and Sons.

Hix, D., Swan II, J.E., Gabbard, J.L., McGee, M., Durbin, J., and King, T. (1999), "User-Centered Design and Evaluation of a Real-time Battlefield Visualization Virtual Environment," *IEEE Virtual Reality*, 96-103.

Hodges, L.F., Kooper, R., Meyer, T.C., Rothbaum, B.O., Opayke, D., de Graaff, J.J., Willford, J.S., and North, M.M. (1995), "Virtual Environments for Treating the Fear of Heights," *IEEE Computer*, 28(7): 27-34.

Howard, I.P. (1993), "Spatial Vision Within Egocentric and Exocentric Frames of Reference," *In S.R. Ellis (Ed.), Pictorial Communication in Virtual and Real Environments*, London: Taylor & Francis.

[Http1] "Elumens Visualize Bright Ideas: SPI API for Performer Version 1.0," (Aug. 8, 2001), http://www.elumens.com/.

[Http2] http://www.ascension-tech.com/products/laserbirds (Nov. 10, 2001).

[Http3] http://www.sgi.com/software/software.html#IRIX.

Jacoby, R. and Ellis, S. (1992), "Using Virtual Menus in a Virtual Environment," S*IGGRAPH 1993 Course Notes No. 40*.

Kallman, E. (1993), "My Anxieties about Virtual Environments," *Presence*, 2(2): 141-142.

Kallmann, M. and Thalmann, D. (1999), "Direct 3D Interaction with Smart Objects," *Proc. of the Virtual Reality Annual International Symposium (VRAIS'99)*, 124-130.

Kaur, K., Maiden, N., and Sutcliffe, A. (1996), "Design Practice and Usability Problems with Virtual Environments," *Proc. of the Virtual Reality World 96*, 34-39.

Kennedy, R.S., Lane, N.E., Berbaum, K.S., and Lilienthal, M.G. (1993), "Simulator Sickness Questionnaire (SSQ): An Enhanced Method for Quantifying Simulator Sickness," *Int'l Journal of Aviation Psychology*, 3(3): 203-220.

Kennedy, R.S. and Lilienthal, M.G. (1995), "Implication of Balance Disturbances Following Exposure to Virtual Reality Systems," *Proc. of the IEEE Virtual Reality Annual International Symposium (VRAIS'95)*, 35-39.

Kennedy, R.S., Jones, M.B., Lilienthal, M.G., and Harm, D.L. (1993), "Profile Analysis of After Effects Experienced during Exposure to Several Virtual Reality Environments," *AGARD Conference Proc. 541 – Virtual Interfaces: Research and Applications, 2-1-2-9*.

Kennedy, R.S. and Stanney, K.M. (1996), "Postural Instability Induced by Virtual Reality Exposure: Development of a Certification Protocol," *International Journal of Human-Computer Interaction*, 8:25-47.

Kessler, G., Bowman, D., and Hodges, L. (2000), "The Simple Virtual Environment Library: An Extensible Framework for Building VE Applications," *Presence*, 9(2): 187-208.

Kim, W.S., Tendick, F., and Stark, L. (1993), "Visual Enhancements in Pick-and-Place Tasks: Human Operators Controlling a Simulated Cylindrical Manipulator," *In S.R. Ellis (Ed.), Pictorial Communication in Virtual and Real Environments*, London: Taylor & Francis.

Kozak, J.J., Hancock, P.A., Atthur, E.J., and Chrysler, S.T. (1993), "Transfer of Training from Virtual Reality," *Ergon*, 36:777-784.

Kruger, W., Bohn, C.A., Frohlich, B., Schuth, H., Strauss, W., and Wesche, G. (1995), "The Responsive Workbench: A Virtual Work Environment", *IEEE Computer*, 28(7): 42-48.

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

Lantz, E., Bryson, S., Zeltzer, D., Bolar, M.T., de La Chapelle, B., and Bennett, D. (1996), "The Future of Virtual Reality: Head Mounted Displays Versus Spatially Immersive Displays," *SIGGRAPH 1996 Conference Proceedings*, 485-486.

LaViola, J.J., Jr. (2000), "MSVT: A Virtual Reality-Based Multimodal Scientific Visualization Tool," *Proc. of the 3rd IASTED International Conference on Computer Graphics and Imaging*, 1-7.

Lewis, C.H. and Griffin, M.J. (1997), "Human Factors Consideration in Clinical Applications of Virtual Reality, " *In G. Riva (Ed.), Virtual Reality in Neuro-Psycho-Physiology Amsterdam*, IOS Press, 35-56.

Liang, J. and Green, M. (1994), "JDCAD: A Highly Interactive 3D Modeling System," *Proc. of the Computer Aided Design and Computer Graphics, Computer and Graphics*, 499-506.

Lin, C., Loftin, R.B., and Nelson, R.H. (2000), "Interaction with Geosciences Data in an Immersive Environment," *Proc. of the IEEE Virtual Reality*, 100-104.

Lin, C., Loftin, R.B., Leiss, E.L., and Su, Simon (2001), "Interaction with Medical Volume Data," *The 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI2001)*.

Loftin, R.B. and Kenney, P. (1995), "Training the Hubble Space Telescope Flight Team," *IEEE Computer Graphics and Applications*, 15(5): 31-37.

McDowall, I.E., Bolas, M., Pieper, S., Fisher, S.S., and Humphries, J. (1993), "Implementation and Integration of Counterbalanced CRT-Based Stereoscopic Display for Interactive Viewpoint Control in Virtual Environments Applications," *Proc. of the SPIE 1256-16*.

Mine, M. (1995), "Virtual Environment Interaction Techniques," Tech. Report, TR95-018, UNC.

Mine, M., Brooks, F., and Sequin, C. (1997), "Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction," *SIGGRAPH 1997 Conference Proceedings*, 19-26.

Mine, M. (1997), "ISAAC: A Meta-CAD System for Virtual Environments," *Computer-Aided Design*, 29(8): 547-553.

Meyer, K. and Applewhite, H. (1992), "A Survey of Position Trackers," *Presence*, 1(2): 173-200.

Moline, J. (1997), "Virtual Reality for Health Care: A Survey", *In G. Riva (Ed.), Virtual Reality for Neuro-Psycho-Physiology: Cognitive, Clinical and Methodological Issues in Assessment and Rehabilitation*, Amsterdam: IOS Press.

[NASA] Chen, J. (2001), "A Virtual Environment System for the Comparative Study of Dome and HMD, Software User's Guide (Version 1.0)", August 5, 2001.

Pausch, R., Crea, T., and Conway, M. (1992), "A Literature Survey for Virtual Environments: Military Flight Simulators Visual Systems and Simulator Sickness," *Presence*, 1(3): 344-363.

Pausch, R., Shackelford, M.A., and Proffitt, D. (1993), "A User Study Comparing Head-mounted and Stationary Display," *Proc. of the IEEE Symposium on Research Frontiers in Virtual Reality*, 41-45.

Pausch, R., Brockway, D.T., and Weiblen, M. (1995), "Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures," *SIGGRAPH 1995 Conference Proceedings*, 399-400.

Pausch, R., Proffitt, D. and Williams, G. (1997), "Quantifying Immersion in Virtual Reality," *SIGGRAPH 1997 Conference Proceedings*, 13-18.

Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R., and Mine, M. (1997), "Image Plane Interaction Techniques in 3D Immersive Environments," *Proc. of the ACM Symposium on Interactive 3D Graphics*, 39-44.

Pierce, J., Stearns, B., and Pausch, R. (1999), "Voodoo Dolls: Seamless Interaction at Multiple Scales in Virtual Environments," *Proc. of the ACM Symposium on Interactive 3D Graphics*, 141-146.

Platt, P.A., Dahn, D.A., and Amburn, P. (1991), "Low-Cost Approaches to Virtual Flight Simulation," *Proc. of the IEEE 1991 National Aerospace and Electronics Conference (NAECON)*, 23: 940-946.

Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. (1996), "The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR," *Proc. of the ACM Symposium on User Interface Software and Technology (UIST'96)*, 79-80.

Poupyrev, I., Weghorst, S., Billinghurst, M., and Ichikawa, T. (1997), "A Frame Work and Testbed for Studying Manipulation Techniques for Immersive VR," *Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST'97)*, 21-28.

Poupyrev, I., Tomokazu, N., and Weghorst, S. (1998), "Virtual Notepad: Handwriting in Immersive VR," *Proc. of VRAIS'98*, 126-132.

Poupyrev, I., Weghorst, S., Billingburst, M., and Ichikawa, T. (1998), "Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques," *Computer Graphics Forum*, 17(3): 41-52.

Pratt, D.R., Zyda, M., and Kelleher, K. (1995), "Virtual Reality: In the Mind of the Beholder," *IEEE Computer*, 28(7): 17-19.

Raskar, R., Brown, M.S., Yang, R., Chen, W., Welch, G., Towles, H., Seales, B., and Fuchs, H. (1999), "Multi-Projector Displays Using Camera-Based Registration," *Proc. IEEE Visualization'99*.

Richard, P., Birebent, G., Coiffet, P., Burdea, G., Gomex, D., and Langrana, N. (1996), "Effect of Frame Rate and Force Feedback on Virtual Object Manipulation," *Presence*, 5(1): 95-108.

Robinett, W. and Holloway, R. (1992), "Implementation of Flying, Scaling, and Grabbing in Virtual Worlds," *Proc. of the ACM Symposium on Interactive 3D graphics*, 189-192.

Robinett, W. and Rolland, J.P. (1992), "A Computational Model for the Stereoscopic Optics of a Head-Mounted Display," *Presence*, 1(1): 45-62.

Rohlf, J. and Helman, J. (1994), "IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics," *SIGGRAPH 1994 Conference Proceedings*, 100-104.

Rosenthal, M., State, A., Lee, J., Hirota, G., Ackerman, J., Keller, K., Pisano, E.D., Jiroutek, M., Muller, K., and Fuchs, H. (2001), "Augmented Reality Guidance for Needle Biopsies: A Randomized, Controlled Trial in Phantoms," *Proc. of the Medical Image Computing and Computer-Assisted Intervention*, 100-104.

Shneiderman, B. (1998), "Designing the User Interface: Strategies for Effective Human-Computer Interaction," MA: Addison-Wesley.

[SGI] "IRIS Performer™ Programmer's Guide," SGI Document No. 007-1680-040. 1999.

Slater, M., Steed, A., and Usoh, M. (1993), "The Virtual Treadmill: A Naturalistic Metaphor for Navigation in Immersive Virtual Environments," *Proc. of the 1st Eurographics Workshop on Virtual Reality*, 71-86.

Slater, M., Usoh, M., and Steed, A. (1995), "Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality," *ACM Transactions on Computer-Human Interaction*, 2(3): 201-219.

Stanney, K.M., Mourant, R.R., and Kennedy, R.S. (1998), "Human Factors Issues in Virtual Environments: A Review of the Literature," *Presence*, 7(4): 327-351.

Stoakley, R., Conway, M., and Pausch, R. (1995), "Virtual Reality on a WIM: Interactive Worlds in Miniature," *Proc. of the ACM CHI'95*, 265-272.

Sutherland, I.E. (1965), "The Ultimate Display," *Proc. of the IFIP Congress*, 2: 506-508.

Sutherland, I.E. (1968), “A Head-Mounted Three-Dimensional Display,” *Proc. of the Fall Joint Computer Conference,* AFIPS, 33: 757-764.

Swartz, K., Thakkar, U., Hix, D., and Brady, R. (1999), “Evaluating the Usability of Crumbs: A Case Study of VE Usability Engineering,” *Proc. of the IEEE IPT’ 99*, 100-104.

Tate, D.L., Sibert, L., and King, T. (1998), “Virtual Environments for Shipboard Firefighting Training,” *Proc. of the Virtual Reality Annual International Symposium (VRAIS’98),* 61-68.

Teitel, M.A. (1990), “The Eyephone: A Head-Mounted Stereo Display,” *Proc. of the SPIE,* 1256-20: 168-171.

Tsingos, N., Funkhouser, T., Ngan, A., and Carlbom, I. (2001), “Modeling Acoustics in Virtual Environments Using the Uniform Theory of Diffraction,” *ACM SIGGRAPH 2001Conference Proceedings*, 545-552.

Usoh, M., Arthur, K., Whitton, M.C., Bastos, R., Steed, A., Slater, M., and Brooks, F.P. (1999), “ Walking > Walking-in-Place > Flying, in Virtual Environments,” *ACM SIGGRAPH 1999 Conference Proceedings*, 359-364.

van Dam, A., Forsberg, A.S., Laidlaw, D.H., LaViola, J.J., Jr., and Simpson, R.M. (2000), “Immersive VR for Scientific Visualization: A Progress Report,” *IEEE Computer Graphics and Application*, 20(6): 26-52.

Vince, J. (1995), “*Virtual Reality Systems*,” Addison-Wesley.

Wloka, M. and Greenfield, E. (1995), “The Virtual Tricorder: A Uniform Interface for Virtual Reality,” *Proc. of UIST’95,* 39-40.

Ware, C. and Osborne, S. (1990), “Exploration and Virtual Camera Control in Virtual Three Dimensional Environments,” *Computer Graphics,* 24(2): 175-183.

Ware, C. and Balakrishnan, R. (1994), “Reaching for Objects in VR Displays: Lag and Frame Rate,” *ACM Transactions on Computer-Human Interaction*, 1(4): 331-356.

Welch, G., Bishop, G., Vicci, L., Brumback, S., and Keller, K. (2001), “High-Performance Wide-Area Optical Tracking: The HiBall Tracking System,” *Presence,* 10(1): 1-22.

Wilson, J.R. (1997), “Virtual Environments and Ergonomics: Needs and Opportunities,” *Ergonomics*, 40:1057-77.

Zhai, S., Buxton, W., and Milgram, P. (1994), “The “Silk Cursor”: Investigating Transparency for 3D Target Acquisition,” *Proc of CHI’94*, 459-464.

Zhai, S. and Senders, J.W. (1997), “Investigating Coordination in Multidegree of Freedom,” *Proc. of the Human Factors and Ergonomics Society 41st Annual Meeting,* 1249-1262.

Appendix A User's Guide

In this section, we describe the method of integrating an existing database into this testbed. We have another guide [NASA] which has a complete explanation of the overall system. It goes through every interface and lists all operations the operator needs to do before, during, and after subject exposure. The guide includes system setup (cable connections, testing, and display format setup), detailed explanations of interfaces (GUIs and commands), basic Unix operations (for the operator who does not know Unix), and trouble-shooting. Here, we present details that are more technical and useful for further development of this software. Thus, this guide is much more written toward developers, while the User's Guide for NASA is written toward end-users.

Loading a New Visual Database

(1) Import an ASE model file

The ASE format file defines a number of tokens, such as object name, vertex, normal, material, texture, light, and other information in a scene database. When programming the ASE scene generator in conjunction with the OpenGL Performer library, we should notice that the XZY orientation of the ASE data for vertices and normals may create problems when displaying the models. The main problem is that the model will not display correctly if working on a XYZ coordinate system. What we did was swapping the Z and Y values of vertex and normal data. In the ASE file, negative Z values go out of the screen. For our applications, we have negative Z goes into the screen. When loading the ASE file, we made all positive Z values negative and all negative Z values positive, and the problem was gone. This was also done for vertex data and normal data.

(2) Specify the geometry of a subject

The geometry of the subject (also called player) is defined by a script file. An example of the script file used in the current project is provided below. There is a series of tokens which are supported in this file beginning with star (*). Their usage and the parameters are listed in Table A.1. One can specify these parameters based on the physical size of the subject and the location and size of the scene.

Example player file: player1.plr

```
*initialpos                          100.0 100.0 0
*view                                0.0 0.0 60.0

*model                               "hand.ASE"  "hand"

*isector  0  0  40   0   1   0  5.0    "DHPLAYER_BODY" 1
*isector  0  0  10   0   1   0  5.0    "DHPLAYER_BODY" 2
*isector  0  0  40   0  -1   0  5.0    "DHPLAYER_BODY" 3
*isector  0  0  10  0  -1   0   5.0    "DHPLAYER_BODY"  4
*isector  0  0  40   1   0   0  5.0    "DHPLAYER_BODY" 5
*isector  0  0  10   1   0   0  5.0    "DHPLAYER_BODY" 6
*isector  0  0  40  -1   0   0  5.0    "DHPLAYER_BODY" 7
*isector  0  0  10  -1   0   0  5.0    "DHPLAYER_BODY"    8
*isector  0  0   0   0   1   0  5.0    "DHPLAYER_RHAND" 0
*motion                              5  -5 5 -5  0 0

*comment {
   *input “3D-mouse"                 DHPLAYER_RHAND
   *input "HeadTracker"              DHPLAYER_HEAD
   *input  “Joystick"                DHPLAYER_BODY
}
```

Token	Usage	Comments
*comment	*comment { }	The comment is within braces.
*initialpos	*initialpos x y z	The subject (player) initial position
*view	*view x y z	Offset of head from the body
*model	*model "filename.ASE" "partName"	Body of part by partName is defined by model – filename.ASE
*isector	*isector x y z i j k length part index_for_part	x, y, z is the position i, j, k is the direction vector: i in x, j in y, k in z direction length is the length of the isector. part can be any of these values, DHPLAYER_BODY, DHPLAYER_RHAND, DHPLAYER_LHAND.
*model	*model “name_of_model”	Specifying which ASE file is loaded
*motion	*motion forward back	Set the max speed of the motion model

Table A.1 Tokens and Specifications (1)

(3) Create a new level file

We use a script file, called a level file, to specify the properties of the objects in the scene, for instance, pick-up, collision, and matching object name. We use a series of tokens to indicate these properties. The program defines the behavior of each token. It will parse the script file before running the simulation loop. Before presenting how to build this file, let us look at an example of the script file used in the current project.

Example level file : Level2.lvl

```
*model              "Level2.ASE"

*ground             "Floor"
*ground             "walls"

*path               "Path01"
*path               "Path02"
*path               "Path03"
*path               "Path04"
*path               "Path05"
*path               "Path06"
*path               "Path07"
*path               "Path08"
*path               "Path09"
*path               "Path10"
*path               "Path11"
```

```
*path                           "Path12"
*path                           "Path13"
*path                           "Path14"
*path                           "Path15"

*movable                        "Sphere01" 4
*movable                        "Sphere02" 4
*movable                        "Sphere03" 4
*movable                        "Cylinder01"4
*movable                        "Cylinder02"4
*movable                        "Cylinder03"4
*movable                        "Torus01"4
*movable                        "Torus02"4
*movable                        "Torus03"4
*movable                        "Pyramid01" 4
*movable                        "Pyramid02" 4
*movable                        "Pyramid03" 4
*movable                        "Box01" 4
*movable                        "problembox1" 4
*movable                        "problembox2" 4

*collidable                     "column01" 1
*collidable                     "column02" 1
*collidable                     "column03" 1
*collidable                     "column04" 1
*collidable                     "column05" 1
*collidable                     "column06" 1
*collidable                     "column07" 1
*collidable                     "column08" 1
*collidable                     "column09" 1
*collidable                     "column10" 1
*collidable                     "column11" 1
*collidable                     "column12" 1
*collidable                     "column13" 1
*collidable                     "column14" 1
*collidable                     "column15" 1
*collidable                     "column16" 1
*collidable                     "column17" 1
*collidable                     "column18" 1
*collidable                     "column19" 1
*collidable                     "column20" 1
*collidable                     "column21" 1
*collidable                     "column22" 1
*collidable                     "column23" 1
*collidable                     "column24" 1
*collidable                     "column25" 1
*collidable                     "column26" 1
*collidable                     "column27" 1
*collidable                     "column28" 1
*collidable                     "column29" 1
*collidable                     "column30" 1
*collidable                     "Path01" 1
*collidable                     "Path04" 1
*collidable                     "Path05" 1
*collidable                     "Path08" 1
*collidable                     "Path10" 1
*collidable                     "Path11"1
*collidable                     "Path12" 1
*collidable                     "Path13" 1
*collidable                     "Path14" 1
```

```
*collidable                             "Path15" 1
*collidable                             "Path02" 1
*collidable                             "Path03" 1
*collidable                             "Path06" 1
*collidable                             "Path07" 1
*collidable                             "Path09" 1
*collidable                             "Floor"    1
*collidable                             "walls"    1

*match                                  "Sphere01”  "column27"
*match                                  "Sphere02"  "column26"
*match                                  "Sphere03"  "column25"
*match                                  "Cylinder01" "column20"
*match                                  "Cylinder02" "column19"
*match                                  "Cylinder03" "column21"
*match                                  "Torus01" "column30"
*match                                  "Torus02" "column29"
*match                                  "Torus03" "column28"
*match                                  "Pyramid01" "column23"
*match                                  "Pyramid02" "column24"
*match                                  "Pyramid03" "column22"
*match                                  "Box01" "column16"
*match                                  "problembox2" "column17"
*match                                  "problembox1" "column18"

*comment {
*align "tSphere01" "Sphere01" "alignBox12"
}

*comment {
   In this model:
                                        Path01 = Path09 = Path15
                                        Path02 = Path04 = Path11
                                        Path03 = Path05 = Path12
                                        Path06 = Path08 = Path13
                                        Path07 = Path10 = Path14
}

*objectPath                             "Sphere01" "Path02"
*objectPath                             "Sphere02" "Path04"
*objectPath                             "Sphere03" "Path03"
*objectPath                             "Cylinder01" "Path13"
*objectPath                             "Cylinder02" "Path14"
*objectPath                             "Cylinder03" "Path15"
*objectPath                             "Torus01" "Path01"
*objectPath                             "Torus02" "Path11"
*objectPath                             "Torus03" "Path12"
*objectPath                             "Pyramid01" "Path07"
*objectPath                             "Pyramid02" "Path10"
*objectPath                             "Pyramid03" "Path09"
*objectPath                             "Box01" "Path05"
*objectPath                             "problembox1" ”Path06"
*objectPath                             "problembox2""Path08"

*matchHighLightObj   "Sphere01"         "rockBallPi"
*matchHighLightObj   "Sphere02"         "redBallPic"
*matchHighLightObj   "Sphere03"         "blueBallPi"
*matchHighLightObj   "Cylinder01"       "rockCylPic"
*matchHighLightObj   "Cylinder02"       "redCylPic"
*matchHighLightObj   "Cylinder03"       "blueCylPic"
```

```
*matchHighLightObj   "Torus01"       "rockTorusP"
*matchHighLightObj   "Torus02"       "redTorusPi"
*matchHighLightObj   "Torus03"       "blueTorusP"
*matchHighLightObj   "Pyramid01"   "rockPyrPic"
*matchHighLightObj   "Pyramid02"   "redPyrPic"
*matchHighLightObj   "Pyramid03"   "bluePyrPic"
*matchHighLightObj   "Box01"          "rockBoxPic"
*matchHighLightObj   "problembox1"   "redBoxPic"
*matchHighLightObj   "problembox2"   "blueBoxPic"
```

Note that we also defined a series of tokens whose initial is star (*). These tokens, the usage, and the parameters are listed in Table A.2. One can specify these parameters based on the scene data.

Token	Usage	Comments
*collidable	*collidable “name_of_object" collision_value	The collision_value should be 1 because the body of the player will collide with objects with the first bit (least significant bit) set.
*comment	*comment { }	The comment is within braces.
*ground	*ground “name_of_object”	Indicate the name of the ground object. It is used for defining the border of the scene. The subject can’t go outside of it.
*matchHighLightObj	*matchHighLightObj “name_of_object” “name_of_match_object”	Swapping out the matched object according to which object is selected.
*model	*model “name_of_model”	Specifying which ASE file is loaded
*movable	*movable "nameofobject" collision_value	The collision_value should be 4. This means the right hand is able to pick things up only if the third least significant bit of the collision value is 1.
*objectPath	*objectPath "object"	Swapping out the paths according to which object is selected
*path	*path “name_of_path”	Indicating all the paths loaded

Table A.2 Tokens and Specifications (2)

(4) Create a new application file. (dhApp.app)

```
*player    "player1.plr"
*comment { }
*level       "TheBox01.lvl"  “Level"
```

Tokens are also initialized with star (*). This ‘app’ file only supports three tokens functioning as level, player, and comment. The usage and the parameters are listed in Table A.3. One can specify these parameters based on the filenames defined in the system.

Token	Usage	Comments
*comment	*comment { }	The comment is within braces.
*level	*level “file_name_level” “level_option”	Loading the level file name of file_name_level, and run level as level_option which can have the five options: Level: It is a generic level that shows all the objects visible and enforces alignment of objects to matches generic highlight (wireframe and shaded). Level1: It inherits from "Level" and highlighted objects "pop up" when object is selected. Level2: It inherits from "Level1": moveable objects appear one at a time after object is matched. Level3: It inherits from Level2 and allows one to set the name of an object that is to be used as a path. Level4: It inherits from Level3 and paths that match the current object popup.
*player	*player “file_name_player”	The physical position of subject

Table A.3 Tokens and Specifications (3)