A VIRTUAL ENVIRONMENT SYSTEM FOR THE COMPARISON OF DOME AND HMD SYSTEMS

Jian Chen*, Deborah L. Harm**, R. Bowen Loftin***, Ching-yao Lin*, Ernst L. Leiss*

*Department of Computer Science
University of Houston
Houston, Texas 77204
Tel: 1-713-743-3610
Fax: 1-713-743-3335

***Virginia Modeling Analysis & Simulation Center
Old Dominion University
Suffolk, Virginia 23435

**Neurosciences Laboratory
Human Adaptation and Countermeasures Office (SK3)
NASA/Johnson Space Center
Houston, Texas 77058

{jchen2, chingyao, coscel}@cs.uh.edu, bloftin@odu.edu, dharm@ems.jsc.nasa.gov

Abstract

For effective astronaut training applications, choosing the right display devices to present images is crucial. In order to assess what devices are appropriate, it is important to design a successful virtual environment for a comparison study of the display devices. We present a comprehensive system for the comparison of Dome and HMD systems. In particular, we address interactions techniques and playback environments.

Keywords

Virtual environment system, projection-dome system, head-mounted display system, comparison, pick-and-release tasks, training

1. Introduction

All virtual environments (VEs) are “through the window” systems [1]. Visual feedback is without question the most dominant channel in the overall VE. Various approaches have been implemented, including full-immersive display, spatial-immersive display (SID), and virtual model display (VMD). However, previous studies showed that no uniformly best display exists for all applications [2]. Instead, the goodness of a display is strongly related to the tasks to be performed in a virtual environment [3]. Therefore, it is important to know which device is suitable for which application [4].

The research reported here was conducted at the National Aeronautics and Space Administration (NASA) / Johnson Space Center (JSC). NASA is seeking ways to deliver more effective training while lowering its cost. The use of VE technologies has been proven to be an effective approach [5,6]. This paper presents methods for building a virtual environment system for the comparative evaluation of a projection dome and a head-mounted display for pre-adapting astronauts to micro-gravity. The tasks used in our system were pick-and-release tasks which were assumed as the common tasks executed in training; they can be transferred effectively from a VE to the physical world.

There are three major motivations for this work. First, the dome display is a technology for constructing semi-immersive virtual environments capable of presenting high-resolution images. However, human factors issues related to it are not well understood. Second, neither qualitative studies nor rigorous evaluations of dome systems have been conducted. Third, a dome and a head-mounted display are currently available at NASA/JSC for ground-based training tasks. JSC personnel are eager to learn the merits of each for ground-based training applications in future work. A comparative study is interesting and relevant because the display devices are not equivalent.

To make the environments practical, several fundamental technological problems need to be addressed: 1) choice of display generators and interfaces; 2) configuration of interface devices for interaction with applications; 3) interaction techniques; 4) software structure; and 5) acquisition of performance parameters.

Our goal in this project was to design a virtual environment that addressed all five questions by applying as many VE design principles as possible. We achieved our goal by using the smart object technique, delicate codelets, and a careful choice of interaction techniques. The performance parameters measured in the system include task completion time, task accuracy, and errors.

The rest of the paper is organized as follows: We briefly survey related work in Section 2. Section 3 describes techniques for designing the overall systems. In Section 4, the result system is presented. Finally, in Section 5, we briefly describe some future directions.

2. Related Work

The potential user tasks involved in VE applications are enormous. A thoughtful approach to understanding the tasks is by splitting them into sub-tasks and analyzing the representative small tasks. In fact, interaction tasks can be classified as navigation, selection and manipulations, and system control [7,8]. With respect to the observer, all components can be egocentric or exocentric [9]. We have seen only two formalized evaluations that have been conducted regarding a "search" task. One was done by Bowman and coworkers [10], who compared a Virtual Research VR8® HMD with a CAVE® display. They built a virtual scenario of a corridor. The corridors are textured polygons and no shadow is cast in the scene. Subjects were required to find several well-hidden targets. When designing the two VE systemsfor this scenario, different setups were used for the interaction mode, display mode, main machine, as well as for the systems and libraries for presenting images. Finally, only one performance factor was considered for comparison: task completion time. This research provided guidelines for choosing an appropriate display for a search task. The authors concluded that the physical characteristics of the displays, users' experiences, and the method of doing pilot studies contributed significantly to subject performance. They also observed that HMD was well suited for egocentric tasks.

Pausch and coauthors [11, 12] presented another way for comparing VR displays with a stationary monitor. The targets to be searched for are heavily camouflaged letters. In their system, the head-mounted display is used as a stationary display by fixing the position of the HMD. The subject sits on a chair and holds a tracker in his hand. The two systems have the same resolution, field of view, image quality, and system setup. When executing pilot studies, error, level of fatigue, and search time were counted. The authors concluded that search performance decreased by roughly half when they changed from a stationary display to a HMD. In addition, the subject who wore an HMD reduced task completion time by 23% in later trials with the stationary display.

In conclusion, we see that two comparisons methods have been applied. The first methodology for building a VE was based on an experiment where realistic systems were used and less attention was paid to holding everything constant. In contrast, the second methodology for building an environment was based on an experiment where every factor was held constant in order to maintain the integrity of the statistics. So there exists a dilemma between getting good practical results and a simplified experimental design when designing a VE system. Which method we should employ depends on the applications and results. Since ultimately our system is to be used for training applications, in the first stage of the design, we take the second method, that is, we try to maintain the two systems as similar as possible and minimize the factors that would affect the statistical results.

3. Application Design

3.1 Display Devices

In this study, a Virtual Research VR4® HMD and a customized dome display were used for comparison. The HMD is a lightweight, rugged display with a 1.3" active matrix liquid color display (LCD). The resolution, field of view (FOV), and overlap are 640x480, 60º, and 100%, respectively.

The spherical dome was painted white and serves as a projection surface. The inner surface is 3.7-meter in diameter. It is equipped with an Elumens projector [13] and a motion base (Figure 1). The resolution of this projector is 1280x1024 and the horizontal FOV is 180º. The motion base was not of interest for our study since we try to minimize the differences between the HMD and the dome environments. The subject is seated upright inside the dome during pilot study for both systems.

Obviously, the features of these two systems are quite different in image performance (e.g., brightness, resolution, field of view, contrast ratio), physical space limitation, weight, portability, and cost. In addition to these physical aspects, the interfaces they present to the user also vary. Finally, factors that can induce cybersickness are present in both systems.

3.2 Visual Databases

Our scenarios are similar to those described by Lampton and co-authors [14]. In our experiments, subjects are presented with a virtual room consisting of different colors and shaped objects. The tasks

Figure 1 Projection-Dome Display

require subjects to move the objects on the left side of the room over to matching platforms on the right side of the room.

There are two levels, with different difficulties. Level one is made up of 10298 textured polygons. It represents a room about 21m long by 14m wide with a floor and four walls 1.6m high. On each side of the room are fifteen platforms. Fifteen objects sitting on one side of the platforms are in the shape of a torus, pyramid, cylinder, box, or sphere, combined with color of green, red, or blue. Two obelisks in the middle of the room act as obstacles. Besides the basic geometry shapes, texture, shading, and shadows are drawn. These intrinsic physical properties can provide useful depth information and visual stimuli to the subject.

In level two, the room, objects, and platforms are the same as in level one. However, instead of using obelisks as obstacles, five different shapes of corridors are added. The paths through the corridors have almost the same number of left turns and right turns although the angles of the turns may be different (Figure 2). The model in this level contains 12434 textured polygons. In the pilot study, the subject may either travel without going through a prespecified path or must go through a path that varies with the type of the object picked up.

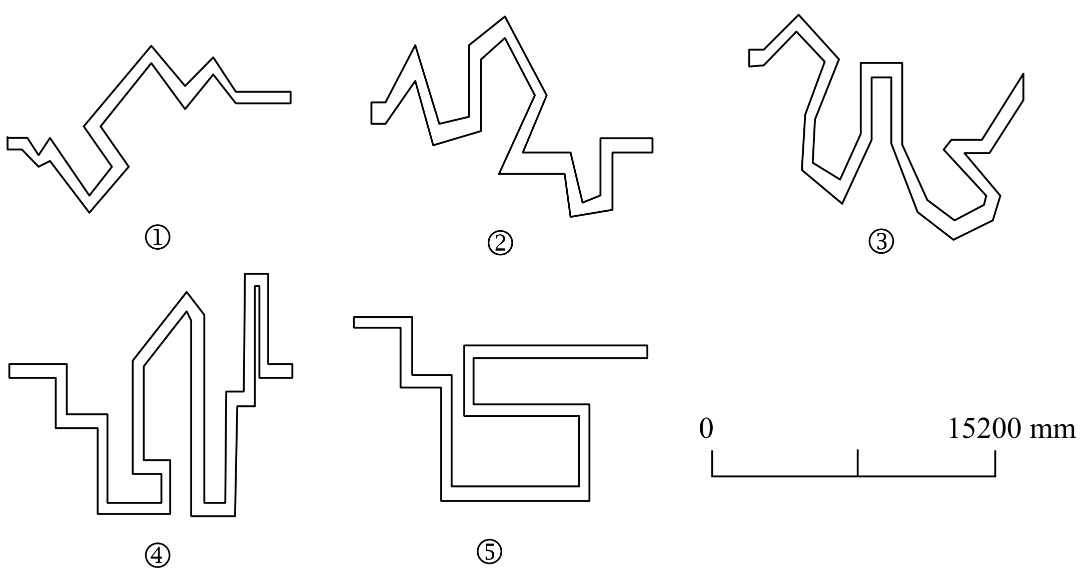

Figure 2 The five Shapes of Paths used in our experiments.

3.3 Interaction Design

Designing successful interaction depends on the task to be performed. Here, we considered the interface for our pick-and-release tasks as the combination of three common types of interactions. These include: body-centric navigation, hand-centric manipulation, and hand-centric selection. In our study, a joystick, a head-tracker (LogiTech™ ultrasonic), and a 3-D mouse (LogiTech™ ultrasonic) are integrated to support interactions.

The *flying vehicle control metaphor* [15] was 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 a joystick, the subject can travel around the scenario. The mappings for both wayfinding and travel are linear. Considering that it is important to implement constraints and limit degrees-of-freedom (DOF) without reducing significantly the user's comfort, we restricted travel to a fixed level relative to the floor of the room. The head tracker, however, is operated in six DOFs. 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. To perform manipulation and selection, the classical *virtual hand metaphor* [16] was employed in this work. An object can be selected by "touch" using a virtual hand. We mapped the scale and position of the subject's physical hand directly to the scale and position of a virtual hand linearly for hand-centric selection and manipulation.

Numerous papers discuss different interaction techniques in virtual environments, such as "put-that-there" [17], flash light technique [18], World-In-Miniature [19], or Scaled World Grab [20]. They were not applied in our first stage of the design, 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 VE is used to duplicate the physical world, not to extend the subject's abilities to perform tasks impossible in the real world.

In order to provide effective visual feedback, an information-rich model was built to make the objects *smart*. Objects are *smart* in terms of their response to the subject's interaction. For example, during exposure, the subject needs to know which object can be picked up. Therefore, an object is wired-framed if it can be picked up (only one object is wired-framed each time); highlighted and wire-framed if it has been picked up; and appears to mark the destination platform on which the object should be deposited. For example, Figure 3 illustrates that a wire-framed green ball is picked; it is also rendered with specular highlights. The path appears once the object is picked, so that the subject knows that he must go through this path to put the ball on the platform on which there is an image. If the ball is dropped in range, the wire-frame, the highlights, and the image will disappear. Another object will be wire-framed, and this process repeats until the subject finishes the current level or runs out of time.

We designed several script files, called *codelets*. Each codelet is given different tokens to define

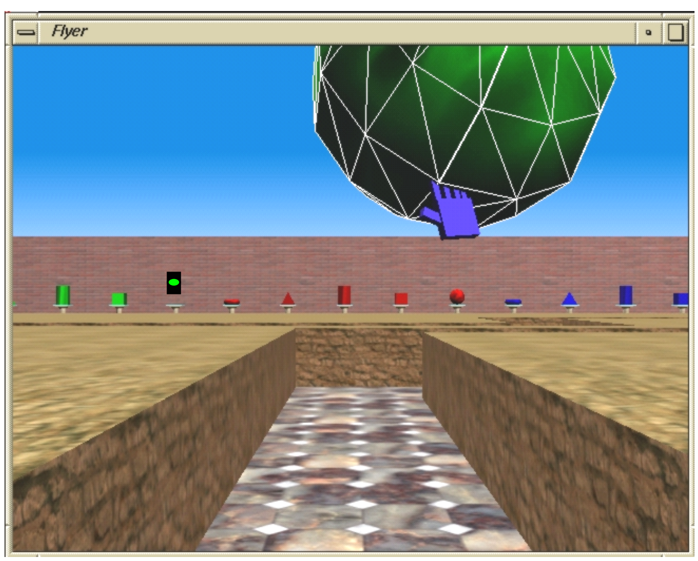

Figure 3 A Screen Shot of the Visual Scenario (Level 2)
The path and the object are *smart* objects.

different behaviors. Without recompiling the program, the operator can specify the visual databases, define objects' behaviors, define interaction metaphors, and input subject data (e.g., hand-head distance, arm distance, etc.). Table 1 lists the tokens used in our codelets. A thesis [21] provides more details regarding their usages. One of the big gains of the codelet is that there is no need to reprogram the whole system if other visual databases are to be loaded or the behaviors of the objects are to be changed. Re-writing a codelet to define the name of the model file and the behaviors (for collision, pickup etc.) will work fine. This should be a convenience for instructors or operators who are not programmers.

Model	Ground	Path
Match	ObjectPath	MatchHighLighObj
Level	Movable	Collidable
Player	Comments	InitialPos

Table 1. Tokens in Codelets

Visual feedback is enhanced by auditory feedback to increase realism. In the system, we have implemented sonification, that is, 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 the trial.

3.4 System Architecture

OpenGL Performer executing on an SGI Onyx® provides the image generation systems. To render correct images on the spherical surface of the dome, we used the Spherical Projection of Image (SPI) Application Program Interface (API) from Elumens [13] to render the distorted images. Since the Onyx does not support auditory output, a PC serves as a sound server. The simulator architecture is illustrated in Figure 4. 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. Since most of them are not programmers, we built an Instructor/Operator Interface (*IOI)*, a graphics user interface (GUI), which provides a virtual platform for the operator. All commands can be issued through IOI by clicking buttons or filling out a form.

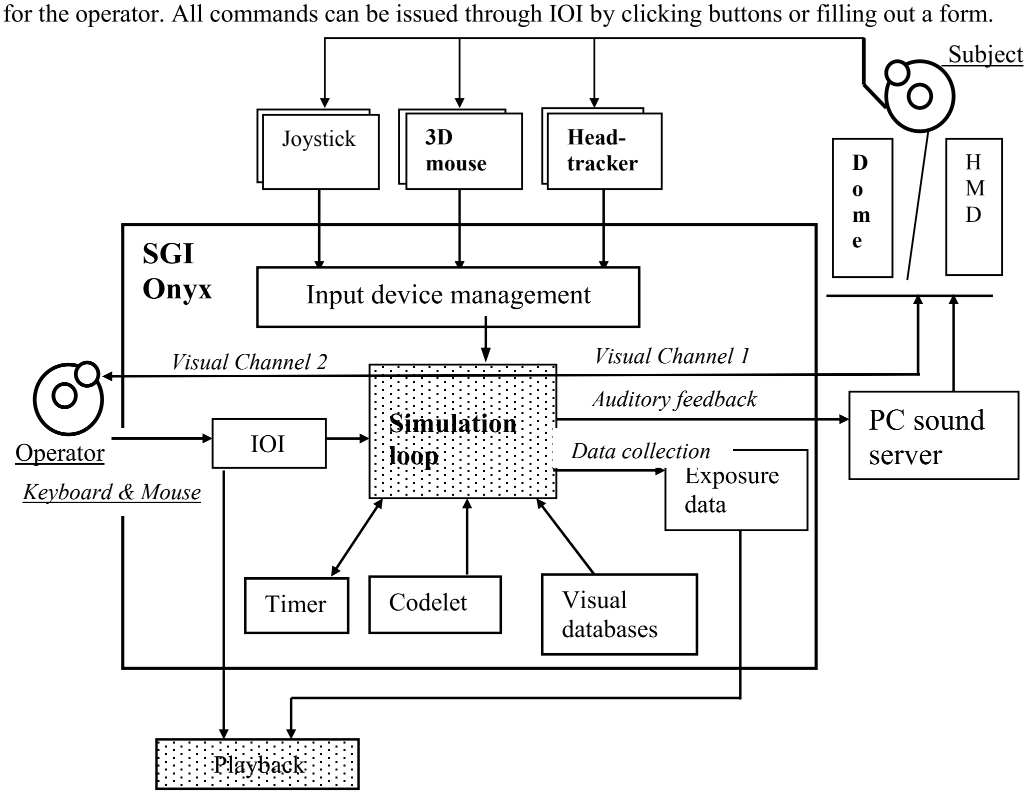

Figure 4 Simulator Architecture

The *simulation loop* is the kernel of our system. It communicates with the other five modules (except the playback module) when running the simulation. The main Performer function parses several predefined codelets, then renders the scene. The loop repeats a series of actions for the duration of the main function. These actions manage the control of the application in each cycle based on the codelets. The simulation loop also captures events from a number 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 subsequently the subject finishes this level in 12 minutes or less, the timer module will leave level one, load level two, and most importantly inform the simulation loop how many minutes remain. If the subject finishes level two in less than 18 minutes, the timer restarts the level one loop. Otherwise, it will terminate this session.

Like other virtual environment applications, our system integrates numerous input devices. The *input device management module* provides an interface to the simulation loop. In 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 and performance data in 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 identification and session name through the IOI and can replay the exposure process of that subject.

3.5 Data Collection and Playback

Traditional data collection is done using videotaping, which has proven useful in postural stability research, or a written account. Subjective reporting using questionnaires is a very common technique [22]. 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.

The data collection module was implemented in software, without interfering with the subject exposure. We simply record every operation, current physical position, and other parameters of interest in an ASCII file. The instructor can open the ASCII file later through IOI for review or playback. This method can be used together with videotape and written documents to investigate subject performance. The data we record include various aspects of the trial, such as task completion time, task accuracy, the subject's current position in each loop, the name of the current picked object, the name of the dropped platform, collision errors, selection errors, and sickness status.

Once the exposure data are collected, we need a way to simulate the exposure process. A 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. The 3D view presents both overviews (global view) and a life-size virtual environment (local view) (Figure 5). This replay duplicates the subject's operations during the run. All these operations can be effected through the IOI by clicking buttons.

4. Result and Conclusion

Figure 6 presents pictures of a pilot study. The left picture demonstrates an HMD test case while the right one demonstrates the Dome test case. In the pilot study, the same physical environment setup was kept for both systems. The only difference in the comparison is the type of the display devices.

One of our purposes when designing the system was to maintain interaction fidelity; therefore we disallowed interaction behaviors that were impossible in the real world. We conclude that the VE system is helpful in measuring the subject's performance and susceptibility to cybersickness, and to study the aftereffects of VE exposure. The system has the potential to serve as a tool with which to build additional evaluation experiments. We are among the first who evaluate the display devices.

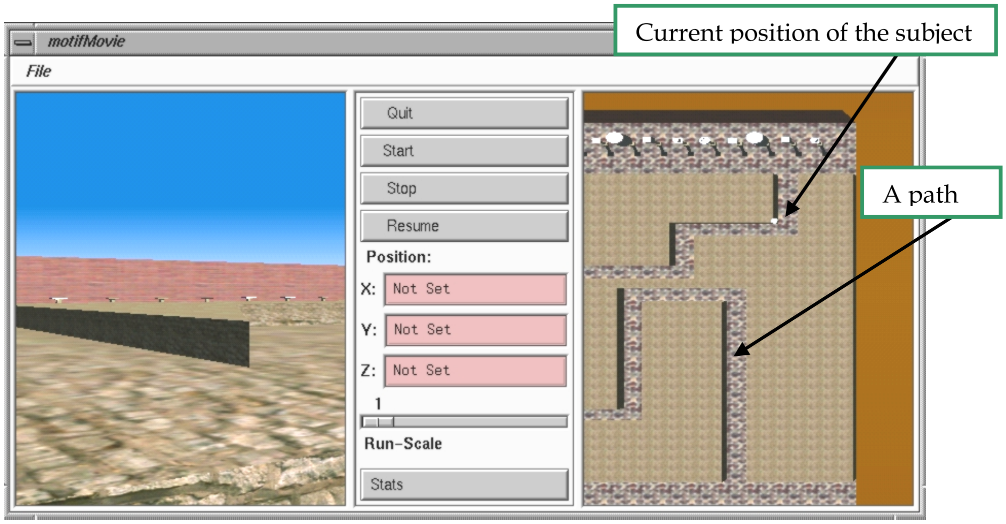

Figure 5 Playback Interface

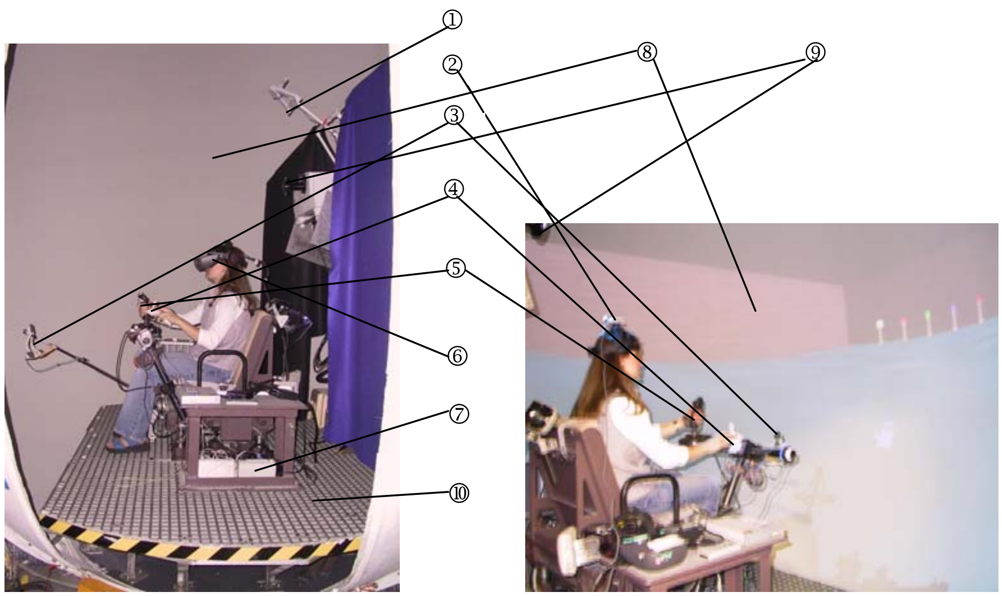

Figure 6 Physical Enviroment Setup of HMD (left) and Dome (right)

① 3-D head-tracker transmitter	⑥ HMD
② 3-D head-tracker receiver	⑦ speaker
③ 3-D mouse transmitter	⑧ spherical projection surface (Dome)
④ 3-D mouse receiver	⑨ Elumens projector (Dome)
⑤ joystick	⑩ motion base (Dome)

5. Future Work

One of the motives driving this work is to extend the current system to a uniform 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.

6. Acknowledgments

This work is supported by NASA NRA-98HEDS-02. We would like to thank Dr. D. Bowman, professor at Virginia Tech, for his answering of queries; C. Cao for setting up an initial version of the system; R. King and H. Garcia, at Old Dominion University, for their technical support and beautiful model creation; and Dr. H.B. Duh, at NASA/JSC, for his comments.

References:

[1] Sutherland, I.E., “The Ultimate Display,” Proc of the IFIP Congress, 65(2): 506-508, 1965.

[2] Brooks, F.P., “What’s Real About Virtual Reality,” *IEEE Computer Graphics & Applications*, 19(6): 16-27, 1999.

[3] Bryson, S., Zeltzer, D., Bolas, M.T., Chapelle, B.D., Bennett, D., and Spitz, E.L., “The Future of Virtual Reality: Head Mounted Displays Versus Spatially Immersive Displays,” *SIGGRPAH’96 Panel Session*, 1996.

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

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

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

[7] Mine, M., “Virtual Environment Interaction Techniques,” *Tech. Reports, TR95-018*, UNC, Department of Computer Science, 1995.

[8] Bowman, D. and Hodges, L.F., “An Evaluation of Techniques for Graphing and Manipulating Remote Objects in Immersive Virtual Environments,” *Proc. of the ACM Symposium on Interactive 3D Graphics*, 35-38, 1997.

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

[10] Bowman, D., Datey, A., Farooq, U., Ryu, Y.S., and Vasnaik, O., “Empirical Comparisons of Virtual Environment Displays,” *Tech. Report TR01-19*, Virginia Polytechnic Institute & State University, Department of Computer Science, VA, 2001.

[11] Pausch, R., Shackelford, M.A., and Proffitt, D., “A User Study Compar-ing Head-mounted and Stationary Display,” *Proc. of the IEEE Symposium on Research Frontiers in Virtual Reality*, 41-45, 1993.

[12] Pausch, R., Proffitt, D., and Williams, G. “Quantifying Immersion in Virtual Reality,” *Proc. of SIGGRAPH’97*, 13-18.

[13] http://www.elumens.com.

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

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

[16] Robinett, W. and Holloway, R., “Implementation of Flying, Scaling, and Grabbing in Virtual Worlds,” *Proc. of the ACM Symposium on Interactive 3D graphics*, 189-192, 1992.

[17] Boleter, R., “Put-that-there”: Voice and Gesture at the Graphics Interface,” *Proc. of SIGGRAPH’80*, 262-270, 1980.

[18] Liang, J. and Green, M., “JDCAD: A Highly Interactive 3D Modeling System,” *Computer & Graphics*, 4(18): 499-506, 1994.

[19] Stoakley, R., Conway, M.J., and Pausch, R., “Virtual Reality on a WIM: Interactive Worlds in Miniature, “ *Proc.of CHI’95*, 265-272, 1995.

[20] Mine, M.R., Frederick, J., Brooks, P., and Sequin, C.H., "Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction," *Proc. of SIGGRAPH'97*, 1997.

[21] Chen, J. "A Virtual Environment System for the Comparative Study of Dome and HMD," *Master Thesis*, *Department of Computer Science, University of Houston*, 2002.

[22] Kennedy, R.S., Jones, M.B., Lilienthal, M.G., and Harm, D.L., "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, 1993.