Tangible User Interface for CAVE^{TM} based on Augmented Reality Technique

JI-SUN KIM

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Denis Gračanin, Committee Chair
Shawn A. Bohner, Committee Member
Mohamed Eltoweissy, Committee Member

December 2005
Blacksburg, Virginia

Keywords: Virtual Reality, Tangible User Interface, Augmented Reality, 3D Interface, CAVE^{TM}

Tangible User Interface for CAVE$^{\mathrm{TM}}$ based on Augmented Reality Technique

JI-SUN KIM

Abstract

This thesis presents a new 3-dimensional (3D) user interface system for a Cave Automated Virtual Environment (CAVE$^{\mathrm{TM}}$) application, based on Virtual Reality (VR), Augmented Reality (AR), and Tangible User Interface (TUI). We explore fundamental 3D interaction tasks with our user interface for the CAVE$^{\mathrm{TM}}$ system. User interface (UI) is comprised of a specific set of components, including input/output devices and interaction techniques. Our approach is based on TUIs using ARToolKit, which is currently the most popular toolkit for use in AR projects. Physical objects (props) are used as input devices instead of any tethered electromagnetic trackers. An off-the-shelf webcam is used to get tracking input data. A unique pattern marker is attached to the prop, which is easily and simply tracked by ARToolKit. Our interface system is developed on CAVE$^{\mathrm{TM}}$ infrastructure, which is a semi-immersive environment. All virtual objects are directly manipulated with props, each of which corresponds to a certain virtual object. To navigate, the user can move the background itself, while virtual objects remain in place. The user can actually feel the prop's movement through the virtual space. Thus, fundamental 3D interaction tasks such as object selection, object manipulation, and navigation are performed with our interface. To feel immersion, the user is allowed to wear stereoscopic glasses with a head tracker. This is the only tethered device for our work. Since our interface is based on tangible input tools, seamless transition between one and two-handed operation is provided. We went through three design phases to achieve better task performance. In the first phase, we conducted the pilot study, focusing on the question whether or not this approach is applicable to 3D immersive environments. After the pilot study, we redesigned props and developed *ARBox*. *ARBox* is used for as interaction space while the CAVE$^{\mathrm{TM}}$ system is only used for display space. In this phase, we also developed interaction techniques for fundamental 3D interaction tasks. Our summative user evaluation was conducted with *ARDesk*, which is redesigned after our formative user evaluation. Two user studies aim to get user feedback and to improve interaction techniques as well as interface tools' design. The results from our user studies show that our interface can be intuitively and naturally applied to 3D immersive environments even though there are still some issues with our system design. This thesis shows that effective interactions in a CAVE$^{\text{TM}}$ system can be generated using AR technique and tangible objects.

Acknowledgments

Most of all, I thank my family,far from here, for always supporting me without question. I could not have finish my thesis without their love and their kind words.

To Dr. Denis Gracanin: I wish to thank my advisor, Denis Gracanin, who is always encouraging and who has helped me to do successful research. He has always supported my research ideas and activities.

To Dr. Shawn Bohner and Dr. Mohamed Eltoweissy: I also would like to express gratefulness to my two committee members. I appreciate their patience in waiting for my final results. Thanks to their many kindnesses, I could concentrate on my research without worrying about the time.

To Patrick Shinpaugh: My special thanks go to Patrick Shinpaugh, who is a System Administrator and Programmer in the University Visualization and Animation Group (UVAG). He helped me out a lot with practical things, such as the use of the VT-CAVE^{TM} system.

To 3DI group members: I would like to thank all members of the 3DI group at Virginia Tech who contributed their efforts either directly or indirectly to the success of this research.

Finally, I want to express my heartfelt gratitude to all of participants who freely devoted their valuable time to my experiments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The purpose of Virtual Reality (VR) is to immerse people in a synthetic world and allow them to interact with computers in a very intuitive way. They are made to feel actually present in the virtual world, where they sense signals (e.g., visual, olfactory or auditory signals) delivered by VR the system. In fact, the goal of the ideal VR system is to have users believe that they are actually doing their tasks in the virtual world. These tasks are composed of fully computer-generated components visually identical to real ones. In addition to the visual components, auditory, olfactory and haptic effects in ideal VR systems provide for real world experiences. For example, in order to simulate flight operations, the ideal VR system provides an experience identical to actual flight operations. The virtual objects ideally respond appropriately to the user's motions. However, current VR systems remain still quite far from the ideal VR system [30]. To experience immersive VR currently, users must put on specialized equipment, such as head mounted displays (HMDs) and tracking gloves. These devices sense and interpret every movement and provide to the VR system the proper input information.

To immerse users in the most realistic virtual scene, three-dimensional (3D) computer graphics is used. If applicable, 3D sound effects generate auditory cues. Thus, VR researchers enable users to interact with the virtual world as if they were really having the simulated experience, thanks to technological inputs and outputs unique to VR and the development of new hardware, software, and interaction techniques.

While traditional human-computer interfaces require that the user sits at a desk and types at a keyboard or presses mouse buttons to interact with the computer, 3D interfaces required for VR enable immersive 3D space where the user works directly with the computer. VR offers means almost impossible with any other technology for users to obtain familiarity with risks both dangerous and expensive.

New concept for 3D user interfaces

The perceived and actual effectiveness of a VR system depends on a large extension on the interface methods. Most computer users are familiar with such 2-dimensional (2D) interface input methods as a keyboard and mouse. These classic input devices are still popular in the Windows, Icons, Menus, and Pointers (WIMP) interface metaphor [17]. And these 2D input devices, for the most part, are still needed for 2D space and some 3D interactions. However, they are not applicable to more immersive 3D environments. Indeed, attempts to adapt traditional 2D interfaces to 3D may create some major problems [41]. Therefore, VR researchers have exploited new concepts for 3D interaction techniques, such as 3D-appropriate menus or 3D-widgets. They have tried to produce a maximum of intuitive and natural user interface, and immersion in virtual environments (VEs), in which users are prohibited for as long as possible from seeing the user interface itself. For that, enhanced interface concepts are required, and consequently many interaction techniques have been developed with new 3D input devices [5].

Until now, relatively little significant research work at all has been performed in 3D interactions for immersive virtual environments, but still the research is ongoing [8]. Most research carried out for desktop systems can be partially applied for new 3D interactions [8]. Currently, many of these ongoing works are very young and need to be more thoroughly explored and tested.

As an approach for 3D UIs, we bring the concept of tangible user interfaces (TUIs) into vision-based 3D user interface, which is unique to a CAVE^{TM} system.

Tangible user interface

The TUI is a more effective and intuitive interface in virtual environments than were earlier interfaces [22]. It enables users to manipulate computer-generated 3D shapes by handling physical objects (props). Similarly, Augmented Reality (AR) systems enable users to interact with virtual objects in the real world [4], and we call it, "Tangible Augmented Reality (TAR)", which refers to TUI's incorporation in AR. TAR interfaces typically require users to wear a see-through HMD to view the augmented real world in an immersive setup [8]. It causes the user to experience seamless space for both real interaction and virtual display in the real world. However, we use physical objects as input tools, not to superimpose virtual images. Tangible interfaces are usually tightly coupled with input devices and the output of user interactions. In our interface, each physical object is mapped to a virtual object, and users can easily handle our input tools, i.e. *props*, to manipulate virtual objects.

A vision-based AR technique is used for this approach to get tracking information. Since virtual images are not directly superimposed onto physical objects, our interface does not augment the real world. Instead, the display space, which displays the output of user interactions, is moved to the CAVE^{TM} screen. Thus, our interface consist of separate spaces for interaction and display. The primary benefit of vision-based tracking system is that the user does not need to carry any cabled or wired input devices.

In our system, all interactions for VR tasks are linked to props' movement. In fact, patterns may allow props to give additional affordances to the user, but they do not augment props themselves. Therefore, the proposed interface system is different from a traditional TAR system because the AR technique does not deliver input information to the separate VR system for interactions.

Fundamental 3D interaction tasks

People desiring to perform tasks in virtual space, from fundamental level tasks to more complex tasks, should experience VR. Complex tasks can be broken down into several lower level steps, such as selection, manipulation (e.g., translation, rotation or scaling), navigation, system control, 3D modeling and symbolic input [8]. These tasks are common and universal in a 3D virtual world.

Particularly, selection and manipulation tasks for virtual objects are basic aspects for 3D interactions, and interaction techniques based on these tasks are also the basis for other complex 3D interaction techniques. Navigation is movement performed by users or the environment itself in and around a virtual environment [8]. System control tasks are required to control system functionalities and change the mode or state of the system. 3D modeling tasks are required for generating 3D geometric objects and manipulating them. The design of 3D interaction techniques should depend on the requirements of these tasks.

Our approach focuses on development of a new 3D interface system brought into the CAVE$^{\text{TM}}$ infrastructure, introducing simple interaction techniques. We will present four interaction techniques for fundamental tasks in Chapter 4, the techniques presented being unique to TUIs using vision-based AR technique.

Two-handed interactions

There has been little research concerning user evaluation of 3D interaction techniques in semi-immersive virtual environments with large displays, such as the virtual workbench and CAVE$^{\text{TM}}$, relative to desktop or HMD configurations [40]. These semi-immersive VEs include actual physical forms of the interaction devices, while the fully immersive VEs can be described metaphorically in terms of the real world. People can handle devices with physical form in semi-immersive VEs because they can see all of the objects in both the real and the virtual worlds. Since people do not need metaphorically to hold physical devices, which are not directly used for interaction techniques, they can use both hands in their main tasks. Based on their tasks, they can choose one-handed operation or two-handed operation with seamless transition. As people usually use two hands to complete tasks, to arrange physical objects and control them in everyday life, two-handed operation in the virtual world is very intuitive and natural.

It is proven by many 3D applications that the physical use of familiar devices, called *tools*, for 3D interactions can improve their usability [8]. Compared to virtual tools, virtual objects with physical tools can be manipulated in direct ways. This direct manipulation can also simplify their controls. Users can be made to feel as if they directly handle virtual objects because they grab physical objects.

For two hands interaction, besides the use of physical objects, data gloves are used as input devices based on gesture and posture recognition of hands. They may be useful to get information about the user's hands and fingers tracked in detail. However, users should memorize many hand-based gestures for various interactions. This results in increasing cognitive load. In addition, with data-glove based interfaces, it is difficult to provide affordances to the user in semi-immersive virtual environments. Users can not only use two hands for manipulating props, but also one hand only, depending on what they want to do in the virtual world. For example, users generally use one hand for navigation task on our interface.

CAVE$^{\text{TM}}$ as an output display

Semi-immersive systems provide a great benefit in many applications, in particular in education, training, architecture and art [2] by allowing simultaneous experience of the physical and the virtual world in the VE, which may not be available with other devices, such as head-mounted displays (HMDs). Originally, the implementation of semi-immersive systems was borrowed from technologies developed for flight simulation. These systems are comprised of relatively high performance graphics computing systems, which include large screen monitors, large screen projector systems, or multiple television projection systems. With a wide field of view (FOV), semi-immersive systems enhance the user's sense of immersion or presence. A CAVE^{TM} system allows the user to see physical objects, even her own body, but everything else is generated by the computer, so that the user can get the feeling of immersion in the VE. Our interface system is developed on the CAVE^{TM} infrastructure, and users can see their physical movement of the props and the output of their interactions simultaneously.

The CAVE^{TM} display provides better resolution and a greater field of view (FOV) than an HMD. In addition, the CAVE^{TM} infrastructure is good for health and safety because the stereoscopic glasses worn by the user and the attached head tracker are much lighter than an HMD, cause less neck strain and make the user feel disclosed to outside the world. When the user is wearing an HMD, the user's eyes have to be very close to something to be focused, and this can cause more eyestrain than when she is using a CAVE^{TM} system.

Current input devices for VEs

To implement 3D interactions in VEs, developers need special devices and equipment to allow the information required for VR applications, such as what users are doing, what they are looking at, and where they are, to be delivered to the VR system, and thereby allow the VR system to present the proper virtual scene to users and respond to users' motions or actions. The data obtained by trackers with various sensors are processed in the VR system to measure the relative position or orientation of one or more objects with respect to each other. Generally, a VR system is equipped with hybrid trackers for more accurate measurement because hybrid trackers can minimize their respective weakness [36]. These trackers usually tether a user with cables attached to the head or other parts of her body. Furthermore, complex movement on the part of the user can cause her to get tangled in cables. Recently, a VR-related hardware company, InterSense [13], has developed wireless tracking modules (i.e., wireless receiver and transmitter pairs) to provide the user with wire-free communication to the virtual world. However, wireless communication is still unreliable compared to wire communication because it has the potential to lose data, which results in latency. Since our current approach is based on the vision technique, our interface is basically wireless. The communication between input system and processing system is directly connected, resulting in minimum data loss.

1.2 Motivation

Computer user interfaces have become more intuitive, directive and natural. Based on human motor perception, the first user interface used text-based input device, such as keyboards. Users usually communicated with the computer by typing on the keyboard. Window-based Graphical User Interface (GUI) provided users a more intuitive interface. In addition, GUIs with pointing devices allowed for touch screen displays, enabling users to interact with the computer by simply touching the screen. People wanted to use their hands with a greater degree of freedom (DOF), and hand-based input devices were developed, such as data gloves. Since tangible interface provide users with physical objects with which to control virtual output, users can interact with the computer more freely and naturally, just as if they perform their tasks with physical objects in the real world. Tangible interfaces provide seamless transition between one-handed and two-handed operation. In addition, tangible input devices are tightly coupled with the output of interactions.

Currently, the most popular input devices in a CAVE^{TM} system are a wand for one-hand operation and data gloves for two-hand operation. Most interaction techniques for fundamental 3D interaction tasks are developed and evaluated often on desktop monitors or HMDs [9]. Although these interaction techniques are developed on same input devices in 3D VEs, their display characteristics are different, for this reason: it is difficult with current interaction techniques to be directly reused in other types of display [9]. In addition, although a wand is a very simply and easily handled, it is only for one-hand operation. Data gloves used for two-hand operation may increase memory load because its interaction techniques are mostly based on hand-gesture or posture. VR researchers are making constant efforts to overcome these drawbacks and steadily develop new interaction techniques.

This thesis proposes a new tangible interface supported by AR technique integrated with a VR system, i.e. the CAVE^{TM} system. The major challenge for this research is to provide a new interface method that people can use in VEs. Most interfaces, popularly used in VEs, require that the user memorize all functionalities. For example, in wand-based interfaces, the user must memorize what functionality is mapped to each button on the wand. In data-glove-based interfaces, she must memorize all of gestures used for interactions. Sometimes, a physical tool is additionally used to provide a menu system and reduces the user's cognition load. However, the user must keep holding the tool in one hand, even when she does not need to access the menu items. Our input devices, i.e. *props*, are very light-weighted and simple shapes, like cards, so that users can grab and hold them as much as they want.

To keep people feeling that they are immersed in the VEs, the interface provides as seamless a way as possible to interact with it. It also supports the user in as intuitive and natural a way as possible. However, to facilitate the designer in developing various but robust interaction techniques in VEs, the interface must provide good affordances in the creation of new interaction techniques by the designer.

Semi-immersive VR systems have grown and their applications become various in different areas, and newly available natural interaction technologies are being continuously challenged to provide use of the auditory or haptic senses. The use of physical objects, i.e. *props*, would not only support natural and intuitive interaction with a 3D virtual world, but it would also help the user's hands move more freely due to the absence of wires or cables. Vision-based AR techniques also help users become unburdened by cables or wires in VEs. Regarding the tracking issue, more specifically the input devices currently dedicated to the CAVE^{TM} system , our approach is not yet completely mature, but rather just a first step toward a new 3D user interface.

The primary goal of this research is to investigate the effect of this new 3D TUI on current CAVE$^{\mathrm{TM}}$ applications and to conduct user evaluations of newly designed interaction techniques for the accomplishment of fundamental 3D interaction tasks. Since this approach is very new, many problems must still be resolved, and its applications studied further (as discussed in Chapter 6).

1.3 Thesis Statement and Organization

The thesis presents a new 3D user interface involving a set of interaction techniques, interface tools, and interaction space. It argues that considerable advantages can be obtained by designing *interface tools* that are specific to the vision technique used for tracking, and *interaction space* for manipulators.

Our approach is new, and the future, through promising, may hold many challenges. The thesis explores the literature on AR, VR, and tangible user interface (TUI), and addresses (in Chapter 2) the mathematical background for 3D coordinate systems .

Our initial system design was very simple, and we conducted the pilot study using this design. To avoid the occlusion of markers, we put the camera over the worktable, so that people or other materials could not interfere with the camera's view. During this pilot study, we observed that the virtual scene was very often frozen due to wireless networking for communication between the AR system and the VR system. We integrated two systems for AR and VR in desktop computer. Also, we attempted to use wired Gigabit Ethernet instead of wireless networking. In both instances, we got quite good performance without any frame freezing or interactive latency. We had to consider where to put the extra light to avoid making a shadow on the worktable; we decided to change the first system design and made an *ARBox*. Another concern that came up was how to restrict the user's movement within the camera's frame to prevent the marker's position being out of the frame because of hand movement. The size of the working area also needed to be considered. The *ARBox* helped to alleviate these issues; however, its design was not quite comfortable enough for the manipulators. After conducting a formative user evaluation, we redesigned *ARDesk*. Both *ARBox* and *ARDesk* are described in Chapter 3.

Additionally, a couple of new interaction techniques should be introduced for object manipulation tasks, like scaling and rotating objects, since it is difficult to directly reuse the existing interaction techniques. After we obtained good feedback from the pilot study, we developed scaling and rotation techniques with additional props. Our system-specific interaction techniques are addressed in Chapter 4.

Formative user evaluation based on the *ARBox* and summative user evaluation based on the *ARDesk* were conducted, and their results are described in Chapter 5.

Finally, Chapter 6 summarizes this study, its contributions to current research, and future directions.

Chapter 2

Related Work and Background

In this chapter, we review the prior work related to the use of TUIs in the performance of fundamental 3D interaction tasks. We first give a general overview of VR and AR, base technologies for our approach, as well as related applications. And then we provide a brief description of base knowledge with respect to coordinate systems.

2.1 Virtual Reality and Augmented Reality

VR systems make users feel as if they are actually inside a completely computer-generated world. In most cases, however, this computer-generated world lacks some features of the real world for the simple reason that the real world is much more complex, so that even the most up to date computing technology cannot generate worlds identical to the real world. For example, features of the real world, such as laws of physics and of material properties, may not be realized in virtual reality systems. There are some libraries that support physics in 3D games, such as Open Dynamic Engine (ODE, http://www.ode.org), but they are not perfect for VEs yet.

Since AR systems work in the real world, they should provide features of the real world, including the features described above. In addition, they should improve the user's task performance in the real world by means of augmented interfaces, such as overlaying 3D images on the real scene. Milgram et al. [35] describes the difference between Virtual Reality and Augmented Reality by presenting a Reality-Virtuality Continuum. The middle region of the continuum represents Mixed Reality (MR), and MR systems implement the augmented world in real and virtual environments respectively. It can be said that represented worlds closest to the Real Environment are mostly real, and that represented worlds closest to the Virtual Environment are mostly virtual. Authors use the term "Augmented Reality" and "Augmented Virtuality" for the MR closest to each end environment. Since AR tries to augment the current real view or environment with virtual objects, users' perceptions could be enhanced by means of AR systems. Since our approach employs AR techniques to get tracking information in a virtual environment, we need to know the characteristics of AR.

In the following sections, we introduce VR and AR technologies and highlight some applications of these technologies.

2.1.1 Virtual Reality (VR)

A specific environment or circumstance can be created by a computer, which can make the user feel actually present in a virtual or simulated world. The more sensory organs involved, the more present the user feels. VR systems utilize special hardware and software to create virtual life experiences.

It is hard to say who first came up with the idea of VR. According to Mazuryk's paper [34], the idea of VR was first addressed in Sutherland's paper in 1965 [42]. But, the term *"Virtual Reality"* was coined by Jaron Lanier, then President of the VPL Research Company, in 1989. Since then, Steve Aukstakalnis and his colleagues have defined VR in their book *Silicon Mirage* as follows: *"Virtual reality refers to experiencing a computer-generated, 3D world, while telepresence refers to experiencing a real but remote environment. "*

It is also hard to classify various VR systems, but P.J. Costello [14] divided most VR systems into three types based on immersion or presence. Although two concepts of immersion and presence are distinguishable, we know both can be regarded as how efficiently VR systems provide users with an effective environment to focus on their tasks. If VR systems that generate immersive virtual environments are utilized, users should consider multiple factors, such as image complexity, stereoscopic view, field of regard (FOR), and the update rate of the display [14]. For example, if the virtual scene contains too many virtual objects or images that are too complicated, immersion should be reduced. Immersion levels can be increased or reduced to affect human sensory organs in various ways. Thus, VR systems adjusted to produce interactive harmony of multiple factors.

Since our interface is designed on a CAVE^{TM} system, which is one of semi-immersive systems, we need to know differences between three types of VR systems. The main three types of immersive virtual environments are Non-Immersive (Desktop) Systems, Semi-Immersive Systems, and Fully Immersive Systems [14]:

- ***Non-Immersive(Desktop) systems***

  Immersive implementation in non-immersive systems is realized in the simplest ways, by such means as the use of high-resolution monitor and/or stereo sound. To interact with non-immersive systems, keyboards, mice or trackball can be used from the desktop. These systems do not require special devices or expensive trackers, allowing users to experience VR inexpensively. However, since non-immersive systems have limitations with regard to the level of immersion provided, when significant immersive implementation is required, these systems are seldom used.

- ***Semi-Immersive Systems***

  Semi-immersive systems were originally implemented in the simulation area. These systems need a relatively high performance graphics computing system because they include a large screen monitor and a large screen projector system or multiple projection systems. Since a large screen provides a wide field of view (FOV), these systems also increase the level of immersion. Even though their cost is considerably high, these systems provide a great benefit, allowing bi-experience of the VE which may not be available with other devices. For example, a CAVE^{TM} system provides users with the ability to see physical objects, including their own bodies. However, everything else is generated by the computer, so that they can feel immersion in the CAVE^{TM} environment. Thus, combination of physical and virtual environments can be a great challenge in a CAVE^{TM} system due to the special characteristics of semi-immersion.

- ***Fully Immersive Systems***

  This type includes the most widely known VR implementation, which forces the user to wear head-worn devices (e.g., HMDs). Everything the user senses, though limited usually to visual or aural stimuli due to their relative popularity, is generated by the computer. Unlike semi-immersive systems, fully immersive systems prevent the user from seeing her own body. Instead, the user experiences virtual objects as representative of the real objects. Since fully immersive systems are blocked from the outside world, these system provide a relatively high level of immersion or presence. However, these systems are also affected by many factors, such as FOV, resolution, refresh rate, and illumination. Typically, HMDs provide lower FOV and resolution than CAVE^{TM} systems.

2.1.2 CAVE^{TM} and Applications

Applications of VR usually produce virtual worlds that contain a lot of 3D models, i.e. virtual objects, with various colors or textures, such as pictures or video images. Navigation in the virtual world is a common task accomplished by a head tracker or a input tracker. Interaction with virtual objects is typically supported by techniques based on specific interfaces with VR systems.

As mentioned in the previous section, a CAVE$^{\text{TM}}$ system can support interactive combinations of both physical and virtual objects. In so doing, VR applications implemented in a CAVE$^{\text{TM}}$ system are quite different from many typical computer applications. Most applications running in a CAVE$^{\text{TM}}$ system are basically highly graphical and interactive. A high resolution display is beneficial to large scale display devices. Its applications are numerous, and they are becoming more diversified.

- **CAVE$^{\text{TM}}$**

  A CAVE$^{\text{TM}}$ system was originally designed in early 1991 by Thomas DeFanti and Dan Sandin and implemented by Carolina Cruz-Neira in late 1991 [16]. The original intent in designing a CAVE$^{\text{TM}}$ system was to overcome poor image resolution, blocking from the real world, and inability to collaboratively work with a group in VEs. CAVE$^{\text{TM}}$ is a projection-based semi-immersive VR system. The computer-generated illusion is displayed onto screen-walls surrounding the viewer. The CAVE$^{\text{TM}}$ system is equipped with head and hand tracking systems to generate the correct stereoscopic perspective scenes. It is also coupled with a sound system to give the user audio feedback and/or increase her immersion by means of stereo effects. The CAVE$^{\text{TM}}$ combines real and virtual objects in the same space without any occluded view of their own bodies when they interact with virtual objects. The user can physically walk around and observe from inside the illusion of being in a computer-generated world, and she can describe and/or otherwise share this experience with others. By these means the CAVE$^{\text{TM}}$ system delivers unique artistic, scientific, entertainment, and educational experiences.

- **Applications**

  Most popular applications using CAVE$^{\text{TM}}$ systems are in scientific visualization, this being CAVE$^{\text{TM}}$'s original intent. Several visualization applications were introduced some time ago [15], for example, architectural walk-troughs, the cosmic explorer, the fractal explorer, regional scale weather displays, and displays of the molecular dynamics of membrane proteins. In early 1990, the only interface for these visualization applications was a wand. The wand input device is very simply operated, allowing the user to press one of several buttons (usually three or four buttons) at once or with two hands to press a button and manipulate a joystick. Although there have been a few input devices developed for the CAVE^{TM} system, the wand is still the most common input device for one-hand operation. For two-hand operation, data gloves are also getting popular.

On the other hand, since the CAVE^{TM} system is room-sized, specially designed vehicles can be used for simulation or training applications [23]. These vehicles are equipped with special interfaces mounted on their bodies. As we mentioned before, the CAVE^{TM} system is well-suited for immersive virtual environments, as in collaborative telepresence [29], surgical training system [31] and 3D artistic medium systems [27].

2.1.3 Augmented Reality

Augmented Reality (AR) is an extension of semi-immersive systems in which the objects manipulated are available simultaneously both in the physical and in the virtual world .

Immersive VR allows the user to experience a completely synthetic virtual environment, but the environment created is still much simpler than the real world since it is difficult to completely realize the real world in a computer as it stands. Moreover, the more realistic the environment, the higher the cost required to create the virtual environment. Unlike VR, AR forces the user to interact with virtual objects in a real environment in real time, and the user can experience more enhanced reality by superimposing the computer generated information, such as visual graphics, sounds, smell or tactical sensations, onto the real world. The scope of applications using AR is widening from stationary to mobile, from standalone to collaborative, and from personal to service environments. A typical AR system contains common components with various technologies combined into a single system. For example, display technologies enable the combination of real and virtual objects into a single view, tracking objects movement and user motion simultaneously. It allows real-time interaction and registration, implementing 3D modeling and calibration. AR systems with these technologies enable users to interact with virtual objects seamlessly in a real world.

Most AR systems have several key issues, such as update rate in real time and the accurate registration of the virtual object with the real scene considered. Since our interface is not on an AR system but uses AR techniques, we need not consider all of the issues that may be of major concern in typical AR systems. Nevertheless, we take a look at common issues in this section because we need to know what technologies should be involved in order to augment the real world and how different our interface system is from a typical AR system.

- **Displays**

  The display unit provides the main output, an augmented real view, and is used to provide a user interface. To represent virtual objects properly in the real scene, AR systems should calibrate the camera with intrinsic and extrinsic parameters. Camera calibration involves the numerical calculation of both types of parameters for each camera. The intrinsic parameters are about how the camera will convert objects within the camera's field of view into an image, and they are independent on position and orientation, while the extrinsic parameters describe the position and orientation of the camera in space. These two parameter sets can be represented as matrices and one 4x3 matrix is obtained as the result of the concatenation. This matrix transforms homogenous world coordinates into homogenous screen space coordinates.

  AR display technology has a few conspicuous characteristics: virtual images displayed are opaque; light entered from the physical world is blocked, virtual light is being used instead; and real images captured are limited by both the resolution of the camera and the resolution of the display.

  In order to view the merged virtual and real images, we will now consider three types of displays, omitting typical desktop monitors.

  1. *Head worn displays*

When users experience AR, they usually wear see-through head-worn displays (HWDs) because these displays provide a seamless means of interaction between output spaces. They are of two types, optical see-through and video see-through. They have relatively low resolution and small field of view, and may not even support stereoscopic view in some models.

2. *Handheld displays*

   As mobile and ubiquitous computing increases, AR is regarded more and more as one of most effective interfaces because a common goal of both ubiquitous computing and AR is to augment the physical real world. In addition, small cameras can be easily inserted into currently available handheld devices, enabling handheld displays to show augmented images, overlaying virtual objects onto the real view grabbed by the camera. Mobile AR is another research area separate from traditional AR. Handheld devices are getting more popular for Mobile AR because of their relatively low cost, compact design, and powerful computing capability.

3. *Projection-based displays*

   The images captured from the camera are projected directly onto physical objects, and people can observe the augmented view by means of a projector or projectors. Two common types of projective display are room-mounted projection and head-worn projection displays. If virtual images are projected onto a room-mounted screen, people may not need to put on any special devices like goggles or HWDs. For example, Raskar et al. [38] introduced the concept of a room-mounted projection-based display into AR technology, although they did not apply this type of display to a specific projection-based VR system, like the CAVE^{TM}.

- **Tracking**

  Typical AR systems require extremely accurate position and orientation of the user in order to align on the display screen virtual objects with the real ones. In fully controlled environments of constrained size, like indoor computing laboratories, researchers have succeeded in tracking the user's movement with high spatial accuracy and resolution, low latency, and high update rates. Those environments provide fairly realistic interaction with computer-generated environments that seemingly coexist with the physical environment. Tracking is defined as measurement of position and orientation of objects in a real world coordination system. The tracking data collected by trackers is used to render 3D images and display virtual objects properly positioned in the video frame. Trackers use different types of sensor data, depending on their different purposes in various environments. Holloway and Lastra [21] addressed the different issues of various tracking systems, with various tracker types.

- **Registration**

  Registration is defined as "*the precise alignment and synchronization of two or more sensory elements*" [3]. In some AR systems, in particular medical applications, registration errors are very crucial, and currently it may be unavoidable to align virtual objects with physical objects precisely. Although tracking technology is a primary concern to the registration issue in AR, data latency can also bring about unexpected registration problems.

  Generally, when we mention registration problems, it seams to have to do with visually aligning in the AR display the virtual image with the real scene. However, there exist other registration problems having to do with auditory, haptic, and even taste or olfactory sensory elements. For example, the user grabs a real tennis racket. When she is about to hit the virtual tennis ball, she should be able to hear the sound and feel the force with which the racket contacts the ball, as well as see the visual alignment of ball and racket. Thus, many factors affect the accurate registration of the virtual with the real.

2.1.4 VR vs. AR

As we have seen above, while *"Virtual Reality"* is based on completely computer-generated environments and focuses on how immersive the user is in the virtual world, *"Augmented Reality"* is based on where the computer generated elements are placed in the real world and how accurate they are registered with the real scene. For example, in immersive VEs you can sit on a real chair while you explore computer generated space, for example the bottom of the sea. With AR systems, you can still sit on the chair that exists in the real world, but you may not feel that you are under the sea, even though you can see virtual fish around you. In AR, the surrounding environment seams real to the user. Thus, goals of VR and AR are different, but combining the two is a challenge to researchers like ourselves.

2.2 Coordinate Systems

A major concern of those wishing to combine real space and virtual space is the incompatibility of different coordinate systems. This manifests itself in the different direction of axes, as well as in the unit of measurement in the virtual world, generated by the CAVE^{TM} system, and the actual tracking information, generated by the AR system. The physical interaction space is used to create tracking information by the AR system for interaction with the virtual world. This tracking information should be differently considered from the tracking information generated by dedicated tracking devices, such as a wand. The tracking information generated by the AR system is based on the camera's frame size, not the CAVE^{TM} room. The approximate ratio of the physical space to the virtual space can be found by calculation. For example, since we use 10 x 10 feet screen walls for display and the unit for trackers used in the CAVE^{TM} room is within the range of -1 and 1 – we call it the *dpf* unit since *dpf* (explained in Chapter 3) is the interface library of DIVERSE,used for our implementation on the VR side, for OpenGL Performer^{TM} –, one unit of *dpf* is matched physically to 1.524m. However, the ARToolKit, which we use as an AR technique in our approach, uses the unit of *mm*. Because of this difference, we need to adjust the position data received from the AR system in CAVE^{TM} applications to manipulate virtual objects and to navigate the virtual world at the proper speed. Even though we could find the ratio by approximate calculation , we have to test our test bed iteratively to get the right ratio. Thus, we need to consider the relationship of the different coordinate systems and units among the AR system, the CAVE^{TM} system, and programming libraries used in both systems. In this section, we describe the coordinate systems for the ARToolKit and CAVE^{TM} and explain how to convert their coordinate systems properly. Based on this knowledge, we will describe our implementation in Chapter 3.

2.2.1 Different Coordinate Systems

- **CAVE^{TM} Coordinate Systems**

  One of the more difficult aspects of programming for CAVE^{TM} applications is dealing with the different coordinate systems. For CAVE^{TM} applications, there are coordinate systems to be considered for the real space, the virtual space, the head, the hand, and each eye. The CAVE^{TM} system involves several different coordinates, and developers would have to devote much more time to solving these considerations were good library support programs not available to them.

  DIVERSE (see Chapter 3), which we choose for our implementation, is commonly used in VT-CAVE^{TM}İt offers *dpfDisplay* class, which contains methods to describe the size, location, orientation, and other aspects of the virtual world. This *dpfDisplay* class makes possible conversions between the physical trackers and the virtual world coordinates based on VT-CAVE^{TM}. The *dpf* coordinate system uses the right-handed coordinate.

- **AR Coordinate Systems**

  In order to realize AR, we must attempt to obtain a relationship between the real and virtual coordinate systems based on data taken from the user's viewpoint. In video-based AR, in fact, the only relationship required is that between the camera coordinates and the 3D world coordinates.

In typical AR systems, certain transformations correlate the coordinate systems of a camera, a virtual object, a world and an augmented image [45]. These transformations are required to merge the virtual objects with the view of the real scene for AR. In our interface system, virtual objects are not displayed in the camera frame, but on the CAVE^{TM} screen walls. Unlike objects in typical AR systems, virtual objects do not need to be aligned with the real scene. However, in order for users to view appropriately the virtual images in the immersive virtual world, these transformations must be performed before illuminating the virtual objects. Three transformations required in our AR system are shown in Figure 2.1.

1. *R*eal-to-Camera (M_{R2C})

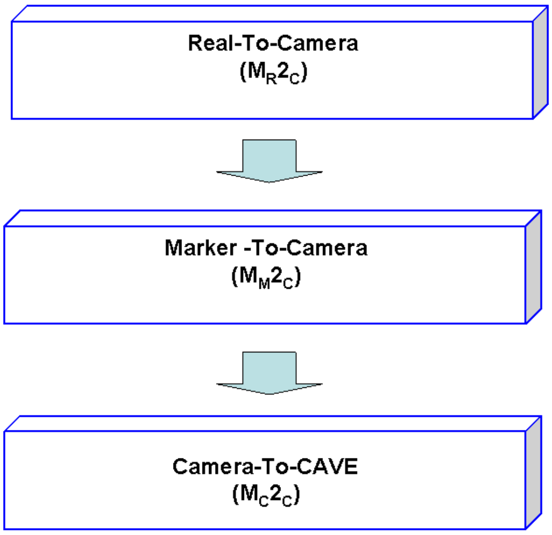


Figure 2.1: AR three transformations

The M_{R2C} transformation specifies that the real world coordinates be converted into the screen coordinates of the video camera. One view frame contains pattern markers and the real scene. The position and orientation of markers within the view frame are used to display virtual objects.

2. Marker-to-Camera (M_{M2C})

   Each marker has its own local coordinate system, and this local coordinate system is adjusted to the screen coordinate system by M_{M2C} transformation. In fact, as shown in Figure 2.1,the augmented real view in the camera frame is the result of the conversion of the local coordinate system of each virtual object to the screen coordinate. However, we do not use the virtual object in our AR system, but instead we only use the original marker's position and orientation data.

3. Camera-to-CAVE (M_{C2C})

   The M_{C2C} transformation produces the final tracking data for the CAVE^{TM} display. In typical AR systems, augmented images are generated for the target display device, such as a monitor or head-worn display, by projection from 3D to 2D display. After the final transformation, the user can see the complete augmented real scene according to the user's eye coordinate system. However, in our system, since the user's eye coordinate system and the camera's coordinate system are perpendicular, we need to adjust this difference before applying the marker's position data to the virtual object in the CAVE^{TM} system. This conversion is also performed by the M_{C2C} transformation

- **Relationship of AR and CAVE^{TM} Coordinate Systems**

  As a result of these transformations in both AR and CAVE^{TM} systems, we can understand the relationship of the coordinate systems to be considered in our interface. Figure 2.2 shows the coordinate systems used in our system. Each number in parentheses indicates the order of rotation about the axis if the Euler rotation method is used.

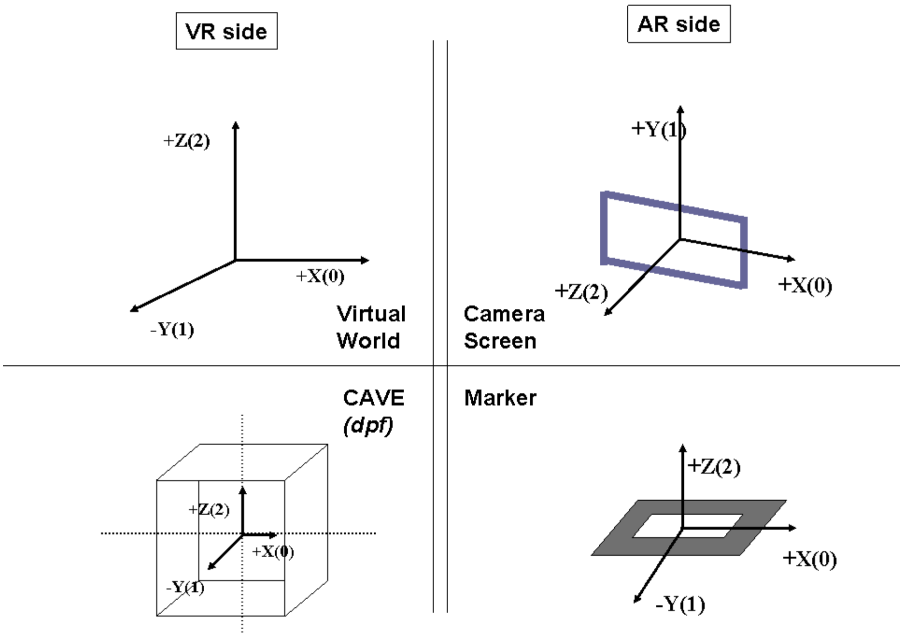


Figure 2.2: Coordinate Systems on VR and AR sides

2.2.2 Camera Properties

In order to get proper position and orientation data regarding pattern markers, we must also consider the camera properties, because proper position and orientation are calculated based on the image frame grabbed by the camera. We look at the camera model, parameters and calibration in this section, using the example of projective geometry to define a transformation matrix for projection of 3D images to a 2D plane.

- **Camera Model**

  Most video cameras used in AR systems are of a perspective or pinhole type [33]. Otherwise, the user would see the augmented view as the same size regardless of distance. The transformation from 3D world coordinates to camera pixel coordinates is performed, and all 3D images within the view frame are projected to the 2D plane, with characteristics of the perspective projection as follows:

1. As Figure 2.3 shows, the view volume is typically shaped like a frustum or truncated pyramid, the viewing frustum consisting of six planes. The frustum's apex is at the eye, and thus the camera, too, has this eye position.
2. Two planes of the frustum are set up perpendicular to the axis. We call these two planes the near and the far plane. Thus, the camera can recognize objects between two planes.
3. The view angle is set by opening the frustum, . The distance between the image plane and the apex of the frustum is referred to as the focal length.
4. All points inside the view volume are projected onto the image plane (see Figure 2.3).

For example, there is some point **P** = [X, Y, Z] inside the frustum, and we can think of the image plane as a 2D image onto which **P** is projected. If the focal length is marked as *f*, then we can have an equation, denoted by

$$x = f\frac{X}{Z}$$
$$y = f\frac{Y}{Z}$$

to establish a point P' = [x, y] on the 2D projection plane.

- **Camera Parameters**

With regards to coordinate systems, two groups of camera parameters influence the process of camera calibration, *intrinsic* parameters and *extrinsic* parameters. In this subsection, we briefly mention these two parameters because the ARToolKit uses a data file for *intrinsic* parameters and its own registration algorithm for *extrinsic* parameters.

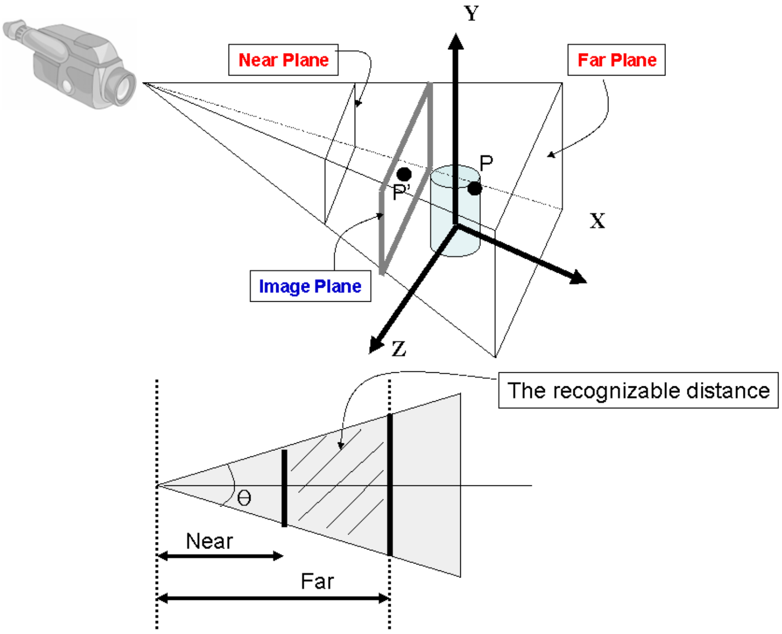


Figure 2.3: 3D View Volume

1. *Intrinsic Parameters*

   The intrinsic parameters are related to the internal geometry of the camera: for example, the focal length, the center position in the pixel image, and the pixel size of the resolution.

2. *Extrinsic Parameters*

   The extrinsic parameters are related to the external properties of the camera, such as camera position and camera orientation. These parameters uniquely identify the transformation between the unknown camera coordinate system and the known world coordinate system [33]. Using these extrinsic parameters, we can define a transformation matrix, which consists of a 3x3 rotation matrix and a 3D translation vector.

2.3 Tangible User Interface

Tangible User Interfaces (TUIs) use physical objects as a means of establishing an interface with the virtual space. They have been studied by many research groups over 20 years [22]. In a natural and intuitive interface, a user can manipulate virtual 3D objects by simply handling physical objects. Our approach is based on a TUI using an AR technique, and is implemented in an immersive VR system, i.e. CAVE$^{\text{TM}}$. Since we chose the ARToolKit, because of its simplicity of implementation, pattern markers had to be attached to the physical objects in order to get the position and orientation information (i.e., pose) needed to manipulate virtual objects. Besides pattern markers, the ARToolKit requires a video camera (or webcam) to capture pose information in real-time. (For details of the ARToolKit, see Chapter 3.) All thus allows the user to interact with virtual objects in an intuitive way by means of physical objects and the AR technology. With respect to using physical objects, the design of our interface contains a few important attributes, such as the availability of one- or two-handed direct operation, the ability to handle multiple objects at once, and the ability to achieve group collaboration [44].

In following sections, we give examples of how TUIs are used for fundamental tasks, such as object selection and manipulation, and navigation in a virtual world. Although our interface system is developed for VR applications, especially for CAVE$^{\text{TM}}$ applications, the base technology used to implement the system is video-based AR. Thus, the examples presented mostly come from ongoing AR systems. Since TUIs in AR systems are referred to *as "TAR" (Tangible Augmented Reality)*, the examples introduced in the following are of *TAR* systems.

2.3.1 Object Selection and Manipulation

Two-handed operations, simultaneous interactions for multiple objects, and group collaboration are common in TAR systems. Generally, TAR applications consists of a table and several physical objects, which are put on the table. Typically, the display device used in

TAR systems is the head mounted display (HMD) because this device enables the illusion, the user experiences, her interactive perspective, to be displayed in front of her eyes. There is no context-switching in these TAR systems. Kato et al. implemented table-top AR environments [25] with conventional markers and paddles for object manipulation. In a similar study, Broll et al. developed Virtual Round Table [10], in which they used actual physical objects, such as mug cups for the collaborative work. In 2003, Kato et al. developed a city-planning system [26] and introduced a CUP interface for virtual object manipulation, such as *pick up*, *put*, *move* and *delete*. And Brown et al. introduced muti-modal environments, i.e., SCAPE combining virtual space and physical space for interior design. They used a physical walking through area, workbench for interactive WIM (worlds in miniature) and HMPD (head mounted projection display) for individual display. Billinghurst et al. [6] showed TAR interfaces for face-to-face collaboration and gave several prototype applications based on their interface design, which can be also applied to computer games [44]. Thus, most of TAR systems with object manipulation functionalities are developed for collaborative work rather than individual work.

2.3.2 Navigation

TUIs have many features, such as two-handed operation and direct and physical interactions, and they have been applied in various application domains to manipulate objects, such as modeling, designing and visualization applications. However, TUIs have been rarely implemented for scene navigation in VEs because this requires specialized interaction techniques in the handling of physical objects. For examples of existing navigational themes, TUIs have just been used to navigate through digital information, such as multimedia stories, virtual museums, virtual music instruments, and scientific 3D data.

Camarata et al. [11] provided a block-based TUI for information navigation installations like information kiosks. In this work, physical blocks, which are electronically augmented with microprocessors, are used to navigate through and explore a virtual gallery. Guzman et al. [19] developed tangible navigational devices for exploring a 3D virtual human body. These various tangible devices resulted in improvement of collaboration and learnability for navigational 3D virtual models. On the other hand, Fjeld et al. [18] presented AR navigation tools for evaluation of scene and viewpoint handling. They showed how a brick-based TUI was used for 3D scene navigation in two views; plane and side views. However, this tool, an improvement of the former "BUILD-IT" system [39], was developed to design scenes that users are planning, not for navigating the whole virtual world. Thus, previous TUI-based navigational interfaces are limited when it comes to traveling virtual spaces.

2.3.3 Others

In addition to the applications mentioned above, tangible interface and AR technology can be used to perform different VR tasks, and we present in this section a few examples of how our interface might be improved in future work for tasks such as system control or 3D modeling.

Occlusion by physical objects, including hands or users' own bodies, can interfere with interactions in vision-based AR technology. However, Lee et al. [28] showed an occlusion-based interaction technique that can be used for system control tasks. In a similar work, Carvalho et al. [12] presented two-hand manipulation using two thumbs. Two pattern markers are mounted on two thumbs, enabling the user to freely control various predefined tasks. Lee et al. [28] employed this occlusion-based manipulation to authoring tools for 3D VR.

On the other hand, an improved approach based on TUI is presented in [1], showing a combination of tangible modeling and graphical interpretation. So far there have been very few researches for 3D modeling tasks in VR. A manipulable physical object is made of small blocks or clay pieces in which computational devices are embedded. For this manipulation, before any interactions take place, the system defines several templates for the virtual objects

.

In addition, TUIs are growing towards pervasive computing interactions [43]. In pervasive computing applications, fiducials in the real world are used as markers for disparate information visualization, so that those applications can simply provide basic interaction techniques for selecting and translating objects. Those fiducial markers are considered natural interaction tools.

The combination of TUI and AR technology can be applied in diverse environments, as these previous works show.

2.4 Summary

In this chapter, we presented related work and mathematical base knowledge. Our interface is implemented on the semi-immersive VR system, i.e. CAVE^{TM}, using AR tracking technology. Since we do not use any electronic or electromagnetic trackers, but rather the video-based AR technique, we addressed the essential components of this AR technology only.

In typical AR systems, the user's ability to experience seamlessly the augmented environment is directly dependent on how robust the registration is. In our approach, we do not consider the accuracy of registration error because our interface system displays virtual images in virtual space, not in the real world. However, we do need to align our physical working space with a virtual world so that virtual objects can accompany the physical movement of the user's hands.

we tried to merge two technologies, VR and AR, bringing in tangible props, to enable the user to explore the virtual space or grab virtual objects, while still looking at the real ones, so that those real objects could give affordances to the user.

Regarding the camera's frustum, the frustum's apex is typically at eye level, whereas the user's eyes are positioned perpendicular to the frustum in our system. Therefore, if the user lifts up props and her two hands get closed to the apex, the actual size of the real interaction space gets smaller. This fact should be taken into consideration when assessing our system design.

Existing TUI-based research describes numerous interaction techniques, some of which we could replicate in our interface. Occlusion-based interaction techniques especially might be employed easily because we do not need to put virtual objects directly onto physical objects. This might be done to improve our current interaction techniques.

Chapter 3

System Design and Implementation

Our interface system has several core components that are used to implement our interfaces for CAVE^{TM} applications. In this chapter, we describe details of how each component is used in the system.

Implementation of our interface system consists of two parts, (1) the VR part for tasks, and (2) the vision (i.e., AR technique) part for tracking. For the VR part, we used the VT-CAVE^{TM} system, which is available at Virginia Tech. We used DIVERSE as a programming library on DADS (DIVERSE Adaptable Display System), which is a cluster system on Linux. For the vision part (from now on *"AR part"*), we used ARToolKit, hand-made props with pattern markers, and a webcam. Our implementation was performed in two different ways.

The first way was to use two separate systems for VR and AR parts. The AR part contains an AR application and a USB-based webcam connected to a laptop. In this design, the tracked data is delivered to the VR part over the TCP/IP network link. Wireless networking was not a good solution for this design because networking issues were significant. Examples include frequent packet loss and view frames frozen because there were static and variable delays between the AR input data and the VR system response. Therefore, we used a gigabit wired-network instead of wireless-network.

The second way was to integrate two separate parts into one computing system. Since ARToolKit only supports a few camera models on the Linux platform, of our several cameras available we decided to use Eyetoy^{TM}. However, due to some time and equipment constraints, we stopped working on implementing the integration design (We describe these constraints in Section 3.2).

The system that we finally adopted is comprised of the following software and hardware components:

- **VR Part**
  1. DIVERSE toolkit(i.e. DTK and DPF).
  2. OpenGL and OpenGL-Performer^{TM}
  3. DADS-based VT-CAVE^{TM}
  4. Stereoscopic glasses with a head tracker
- **AR Part**
  1. ARToolKit
  2. 3Com® HomeConnect® PC Digital WebCam.
- **Proposed interface components**
  1. Props
  2. *ARBox*
  3. *ARDesk*

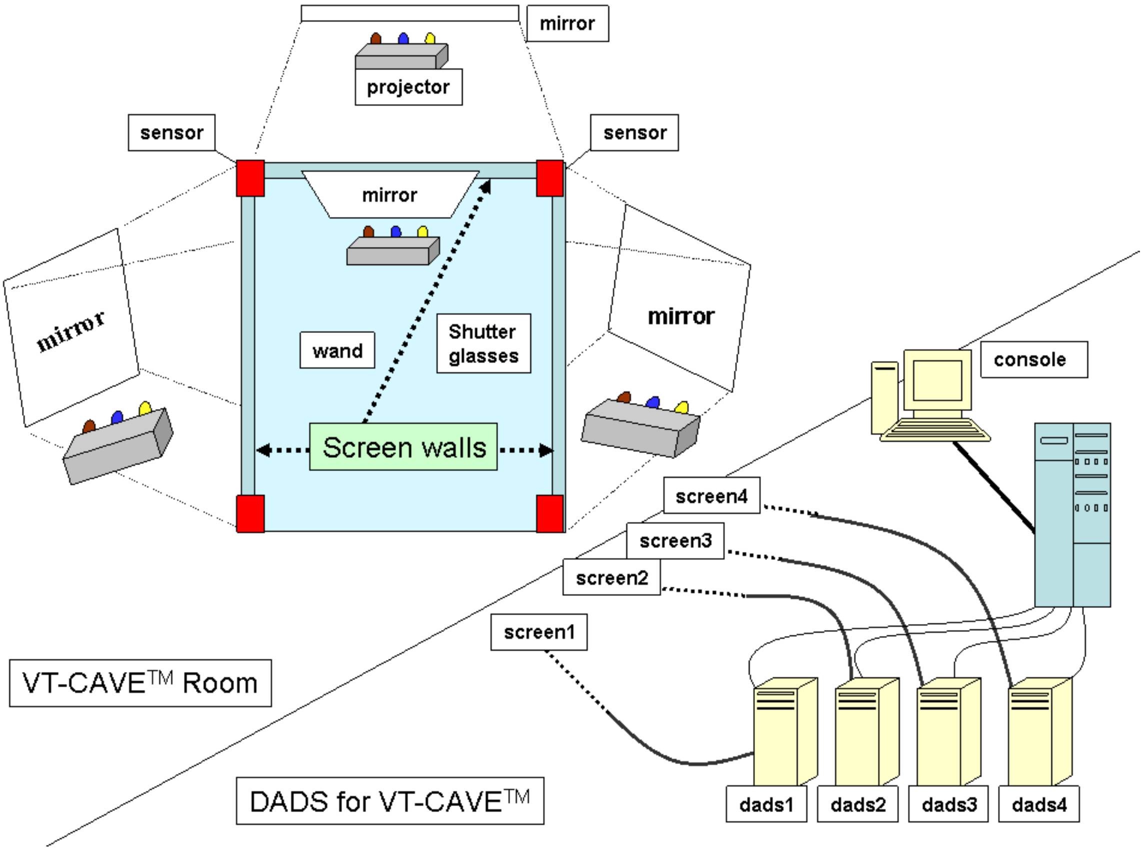


Figure 3.1: VT-CAVE$^{\mathrm{TM}}$

3.1 System Components

3.1.1 VT-CAVE^{TM}

VT-CAVE^{TM} is in the Virtual Reality (VR) laboratory at Virginia Polytechnic Institute and University and has three rear-projected walls and one front-projected floor, each of which is a 10x10 foot projection screen. Stereoscopic images are displayed by a projector for each screen. The CAVE^{TM} infrastructure allows users to be immersed in a VE without heavy HMDs for 3D visualization and interaction with virtual objects. It also enables many individual users, or a group of users, to be immersed together in a computer-generated world. Our interface system is designed on this CAVE^{TM} infrastructure because our goal is to bring a new concept for 3D UIs into the CAVE^{TM} system. In addition, the VT-CAVE^{TM} system can be driven by a Linux-based clustering system. We use this clustering solution, named *DADS*, because it allows driving multiple walls of the CAVE^{TM} system efficiently and inexpensively. We can easily drive the three walls as a main view to display the virtual world and present the user's interaction outputs. The floor screen was to have been used to provide a constant light level within the defined work area in the CAVE^{TM} system; it was not used, however, because the top projector for the floor screen was broken.

- **Glasses**

  To see the projected 3D images correctly and get an immersive 3D experience, viewers have to put on stereoscopic glasses, and only one user can be head-tracked. The glasses alternately *shutter* (i.e., block) the view of the left eye, then of the right eye, preventing each eye in turn from seeing the image, and, therefore, only one eye views the scene at once. The eye can be synchronized with the images displayed on screens because each one has its own blocked vision. Thus, each eye may get a little different image in the virtual world. For example, if the screen has the frame rate as 96 times per second, and each eye would be blocked 48 times a second. Thus, the difference of views watched by each eye makes the user feel depth of view, and consequently the user can experience immersion in VEs.

- **Head Tracker**

  Most CAVE^{TM} applications use the position of a head tracker in real space to correct images from the user's perspective and compensate for her head movement. The tracked data is delivered to the CAVE^{TM} system. Usually, application developers use this data to display the corrected view to the user, since the head tracker is attached to the glasses worn by the user. The user turns her head to look in any desired direction, and while doing so, she can freely control the viewpoint of the virtual scene, and even of virtual objects. For example, the user's viewpoint may be inside the virtual object or around it.

- **Input Device**

  Unlike a fully-immersive VR system, the semi-immersive VR system can allow the user to use graspable input devices or input tools directly without using metaphors to assist input devices because the user can see any physical objects in semi-immersive VEs. This characteristic of semi-immersive VEs is beneficial in that it gives the user comfort and intuitiveness because she can manipulate physical devices or tools without cognitive load, seeing them in the real space. However, currently popular input devices, such as wands, in semi-immersive VEs have controls that are too small to accommodate the many functions required for interaction with the virtual world.

  The origin of our approach came from our desire to replace the dedicated input tracker, i.e. a wand device, with vision-based tangible interface tools. Our first theoretical design included a mock of the wand's functionality. For example, we thought the interface system could be designed with many markers, such as buttons, cursors, and menus, in addition to the dominant props. However, this design did not take advantage of the benefits of TUIs at all because two hands could not be involved in the main task. The current props as input devices were newly designed to take advantage of these benefits. Details are described in Section 3.2 System Design.

3.1.2 DIVERSE and DADS

DIVERSE stands for "Device Independent Virtual Environment that is Reconfigurable Scalable and Extendable," and it was started at Virginia Tech in the late 90's. It is an open source project and provides a cross-platform and API for developing VR applications. The current DIVERSE mainly consists of two parts, DTK (Diverse Toolkit) and DPF (Diverse Interface to Performer$^{\text{TM}}$).

The major functionalities of two libraries are as follows: a common interface to devices, local and remote shared memory, device drivers, a common interface to OpenGL-Performer$^{\text{TM}}$ and a large collection of utilities. The application, running on the CAVE$^{\text{TM}}$ system, uses two libraries together to project interactions with the virtual world onto the walls of the CAVE$^{\text{TM}}$ system. DIVERSE currently operates on both Linux and IRIX platforms, and support for Windows XP and Mac OS X is under development.

DADS (DIVERSE Adaptable Display System) is a clustering solution developed for the lower cost and the more efficient CAVE$^{\text{TM}}$ system, using the DIVERSE API (as its name implies). Since DIVERSE uses *shared memory* on DADS, we do not need to use MPI (Message Passing Interface) or other parallel APIs to drive this clustering system. Although CAVElibs could be used, which are general libraries for the CAVE$^{\text{TM}}$ system, CAVElibs lack some good APIs for non-graphical modules compared to DIVERSE. For example, we can easily control shared memory segments to communicate within DADS or between DADS and other different systems. Developers can generate their own DSOs (Dynamic Shared Objects) to drive their own shared memories, and DSOs work well with other daemons without any conflict. Thus, DADS and DIVERSE used in combination enables us to realize VR systems efficiently and drive CAVE$^{\text{TM}}$ś projection screens easily and at low cost.

We generate our own shared memory to deliver the tracking information to the CAVE$^{\text{TM}}$ application as input data sent from the AR part. To efficiently control this shared memory, we also make our own DSO. This DSO mainly serves for socket communication between the VR part and the AR part. All messages, which include markers' IDs, rotation and scaling information, as well as position and orientation data, are stored in shared memory segments, and the VR application displays this data on screens as the output of interactions.

3.1.3 ARToolKit

The role of the ARToolKit is to analyze every video input frame and generate a transformation matrix for OpenGL to generate an augmented view. This matrix contains the position and rotation information of each marker relative to the camera position. The ARToolKit library, developed by Hirokazu Kato and currently supported at the University of Washington, enables the easy and rapid development of AR applications as it provides the computer vision technique in support of all of image processing, calibration and tracking within a frame. We use this ARToolKit for optical tracking of our tangible interface tools because we do not use the traditional input devices dedicated to the CAVE$^{\mathrm{TM}}$ system. Since this AR technique is based on cheap printed out markers, our interface can be easily integrated with the existing physical objects like wooden blocks. Each marker contains a unique and predefined pattern within a black frame that should be visible during the tracking process by a video camera. Thus, the advantage of ARToolKit is in its simplicity. Although its performance depends on the video camera and computing system involved, currently an off-the-shelf USB webcam and any number of popular, widely available PCs can satisfy the requirement for an updated display rate.

In typical AR applications, although the fast movement of markers might cause the loss of tracking information of objects, it does not create any significant problem because global image processing can cover the whole frame and the user can still see every real scene. Sometimes, any partial occlusion of the marker raises the same problem, the loss of tracking information because ARToolKit cannot detect the marker's pattern when the camera cannot view the complete pattern within the complete black frame. And this problem is significant in our interface system because it makes virtual objects seem to jump as the user watches them.

This difference between our system and other typical AR systems results from the fact that the rendered images displayed to the CAVE^{TM} screens are separated from the camera's view. Most ARToolKit-based applications typically include the following flow:

1. Image capture, with a video camera and image grabber module.
2. Image analysis, with detection of predefined markers
3. Marker recognition, with the threshold image
4. Calculation, of position and orientation of markers relative to the camera
5. Overlaying images, generated by OpenGL, based on the calculation result

For the AR part of our interface system, the last step is excluded, and instead the data obtained in the fourth step are stored in the shared memory segments on the VR system. The shared memory segments are accessed by the CAVE^{TM} application and the virtual world is properly displayed on the CAVE^{TM} screen walls.

3.2 System Design

3.2.1 Preliminary Constraints

Because our interface system is based on the VT-CAVE^{TM} system, we have faced several problems during our research that we could not solve. For examples, the top projector did not work except with a blue lamp. Our original idea was to use the top projector as an extra light, but we had to find another was to provide sufficient brightness of the work area. The VT-CAVE^{TM} system is shared with other research groups at Virginia Tech, and for this reason we could not work continuously on the CAVE^{TM} system and had to make our interaction space movable. In addition, the desktop computer, the only one from which anyone working on an individual research project could access DADS, suddenly died. Sometimes we had to rebuild all of the project files to upgrade the DIVERSE library.

Our design principles were restricted by equipment and time constraints, as well as by the AR technique itself.

3.2.2 Design Principles

Factors	Input devices	Interaction Space
TUI & Usability	Graspable. Light-weighted and comfortable. Visual Affordance.	Non-occlusion of the user's view. Proper height from the floor.
AR & CAVE	Marker's proper size. Markers with strong contrast (i.e. Black & White). Short height of Props.	Limited work area. White-colored working desk. Sufficient brightness.

Table 3.1: Relationship of Factors and Interface

Table 3.1 shows the relationship of factors that affected to our design principles and our interface (i.e., input devices and interaction space). These relationships defined our design principles.

- **Input devices (Props)**

  1. *Graspable*

     Each prop corresponds to a virtual object, and the user may want to manipulate several virtual objects with props at once. In typical TAR (see Chapter 2) systems, most designers use cubes or everyday physical objects like mug cups. The user can grab two props at most with two hands at once. However, we designed card-shaped props with two handles so that the user can grab and handle several props at once with a hand.

2. *Light-weighted and comfortable*

   For the same reasons given above, i.e., to enable the user to grab and handle several props at once, these props should be light-weight and comfortable to handle.

3. *Visual affordance*

   In typical TUI-based applications, the physical object itself, such as a wheel or a cube, gives affordance to the user. Since we decided to use card-shaped props, however, the shape itself provides visual, instead of physical, affordance.

4. *Marker's proper size*

   This has to do with the AR technique. Since we use ARToolKit to get tracking information, the marker's size must be considered. The bigger the size of the marker, the better recognized it will be. However, the smaller the size of the marker, the faster it will be detected. The extent of the trade-off can be determined by means of iterative experiment.

5. *Marker's strong contrast*

   ARToolKit uses the threshold image converted from the real video frame to recognize the marker. To reduce the rate of wrong detection, the marker's pattern should be drawn inside the black square in black against a white background.

6. *Prop's short height*

   We have a very limited work area because all movement must take place within the camera's view frame. This requires that the work area be small. The size of the work area is governed by the distance between the camera and the prop. Tall props end up being too close to the camera. Our prop is card-shaped, its height being about a quarter inch.

- **Interaction space**

1. *Non-occlusion of the user's view*

   People in the CAVE^{TM} room usually walk around with or without an input device to view illusions displayed on the screen. They are not bothered by anything but trackers' wires. Since people are not allowed to walk around using our interface and all of the wires are fixed, people using our interaction space are not annoyed by these wires. However, nothing in the work area should occlude their view.

2. *Proper height from the floor*

   The user's placement within our interaction space is fixed. By turning her head horizontally or vertically she can see every virtual object in the CAVE^{TM} room. If the interaction space were either too low or too high, she would strain her neck and perhaps her arms as well. We must define the proper height of the interaction space to accommodate the user.

3. *Limited work area*

   With respect to the camera's frame size, any of the user's motions that take place outside the camera's view cannot be detected. Therefore, we need to limit the work area to that within the camera's frame. In addition, before the user begins her tasks with our interface, we need to calibrate the work area, taking into consideration the camera's view and the physical dimensions of the interaction space.

4. *White-colored work area*

   ARToolKit is sensitive to the ambient color of the real scene. We found that if the work area, i.e., the desk of our interaction space, was transparent or dark, then recognition performance was decreased. For this reason, we wrapped the desk surface with white paper.

5. *Enough brightness*

   For the video-based AR technique, ambient light around markers must be bright enough so that the camera can recognize them. To achieve this, we attached two clip lamps over the work area of our interaction space. These two lamps provide enough brightness, but they cause shadows, which may cause faulty detection. This remains a problem that we will try to solve in the future.

Based on these design principles, we designed *ARBox* and *ARDesk*. *ARBox* has solved a few issues regarding the positioning of light bulbs, reduction of shadow problems, and restriction of the user's movement within the limited work area. However, during observation of our formative user evaluation, we found that the box environment, especially its blocked space and its size, made users uncomfortable and inhibited the performance of their tasks. After the formative user evaluation, we designed another interaction space utilizing the transparent frames made of plastic glass. In the following sections, we describe our input devices and interaction spaces in details.

3.2.3 Input Devices (Props)

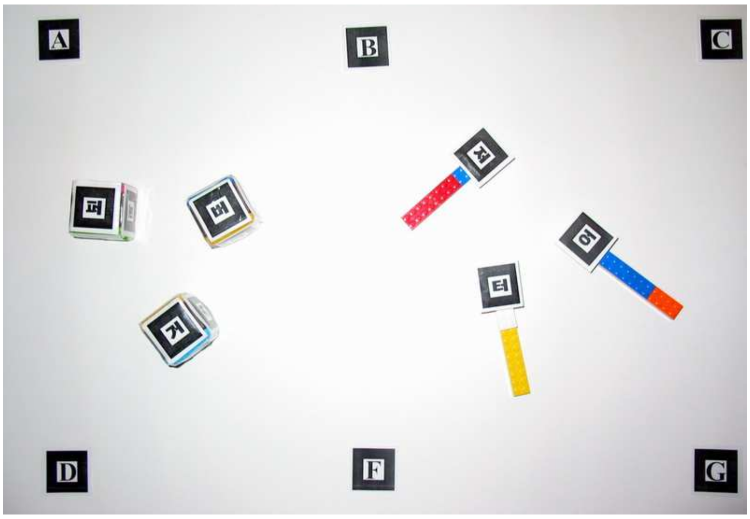


Figure 3.2: Physical Props and White board used for the first pilot study

We designed props over several iterations. Our initial props simply attempted to use cubes – we used alphabet-cube toys for children – or just markers attached to sticks (Figure 3.2).

During the pilot study, we observed that these props could be easily occluded by a user's fingers. Our second design is shown in Figure 3.3.

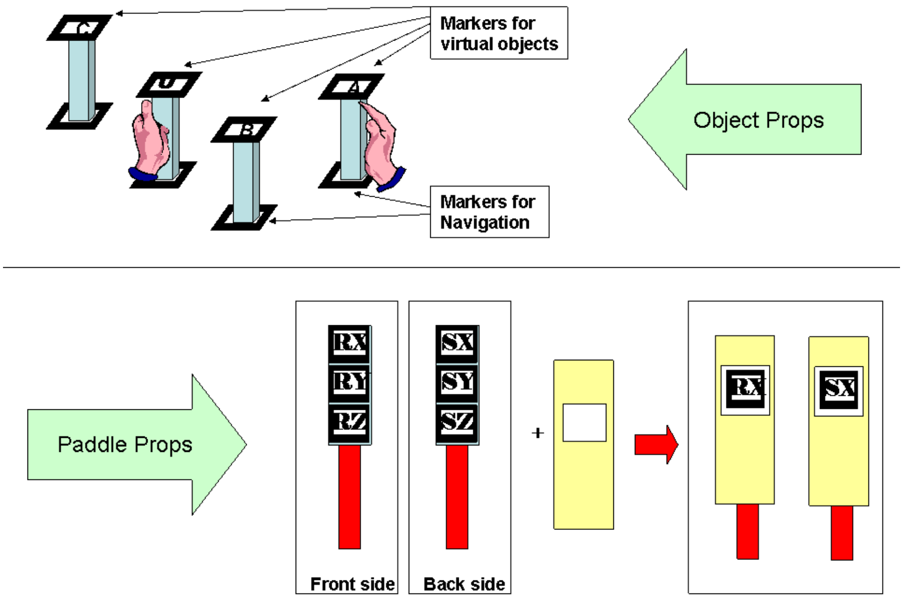


Figure 3.3: The second designed props

Users can easily grab these props without occluding patterns by their finger. We named them *object props*, i.e., props for manipulating objects, and we also designed paddle props for rotation and scaling tasks, which we named *control props*. Users handled these props more freely and easily than they did the earlier ones. However, certain defects still restrict interactions. For example, with the paddle prop, as Figure 3.3 shows, virtual objects can be scaled along only one axis. Object scaling could be performed with the composite axes such as xy, yz, xz or xyz axes. In addition, the object prop's height violated our design principles. We finally redesigned both the object props and the control props.

As Figure 3.4 shows, our object props are formed like cards with two handles. This design is in accordance with our design principles for input devices. With the newly designed control props users can not only show one axis to the camera, but also composite axes, because each marker for a specific axis has its own cover on the control props. Users can open the cover

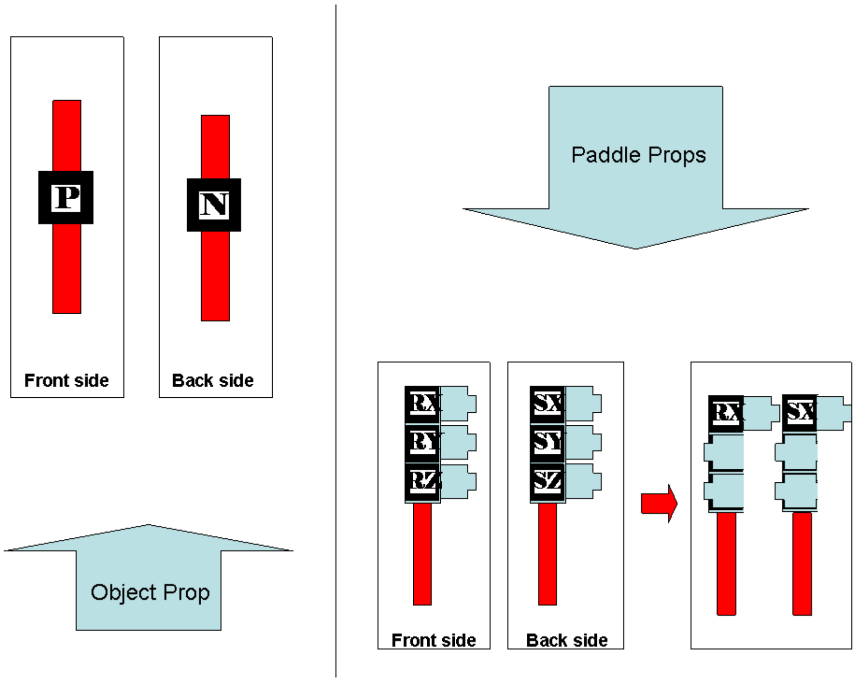


Figure 3.4: The third designed props

to show any axis to the camera.

Figure 3.5 and 3.6 show the current design for our input devices. The current object props provide visual affordance with patterns as well. During the formative user evaluation, we found that users had trouble handling the covers of the control props shown in Figure 3.4. We redesigned these control props, as shown in Figure 3.6, and our summative user evaluation was conducted using this new design.

In fact, one of the results from our summative user evaluation was that our current control props should be changed because users claimed that they could not handle the control props with one hand, even as they had expected. We describe details of user studies in Chapter 5.

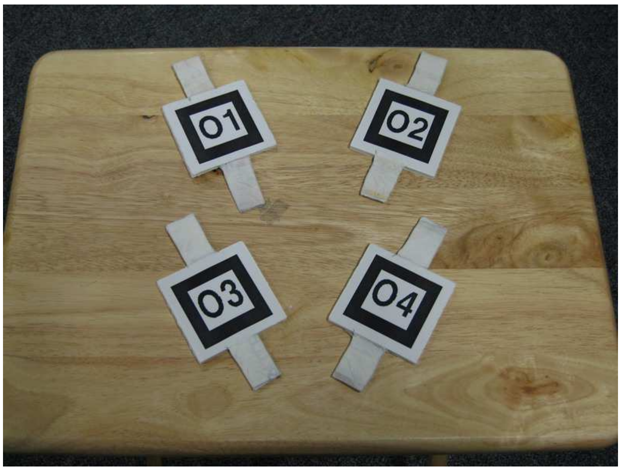


Figure 3.5: The current object props

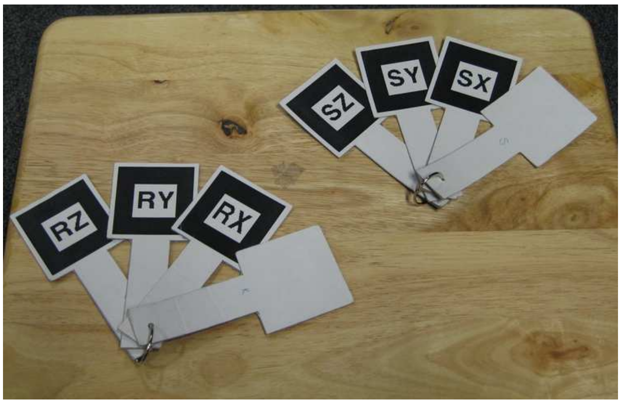


Figure 3.6: The current control props

3.2.4 Interaction Space - *ARBox*

The idea of using a packing box as an interaction space came up due to light and shadow issues. As we have discussed throughout this thesis, the major obstacle to camera recognition of pattern markers is darkness in the CAVE^{TM} room. We first tried placing the extra light bulb over the work desk, as shown in Figure 3.9. We could alleviate the darkness problem using the extra light bulb, but there still exists the shadow of hands or other physical objects, making the recognition worse.

The fact that ARToolKit once detected all black squares as markers within the view frame reduced the system performance. Light bulbs placed in each top corner of the CAVE^{TM} room could have a bad effect on the projected images. Thus, we first decided that we needed to block space for interaction, to camera only views illuminated only by the extra light bulbs. The reason we choose the packing box was that it was cheaper and simpler than making the space out of another material. In addition, reuse of the packing box facilitated development of our interface system.

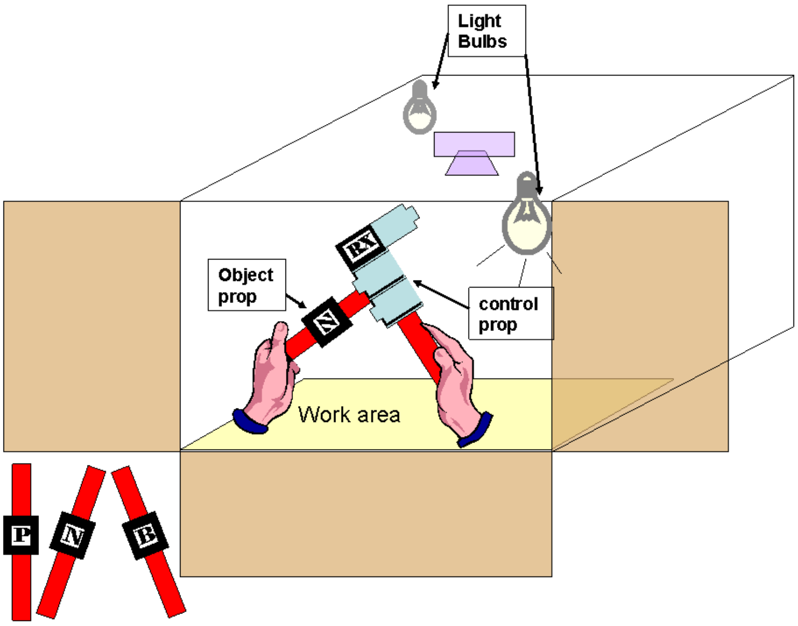


Figure 3.7: The initial ARBox

3.2.5 Interaction Space - *ARDesk*

The final design for our interaction space is illustrated in Figure 3.8. Although *ARBox* performed well with regards to system issues, it came up, with other issues, in the user study. For example, it still hinders the user's vision. In addition, most users wanted to see their own hands during the experiment because they could not understand how they manipulated props in the box, and whether or not props overlapped each other. Since our tracking solution totally depends on the camera, markers on props must not be overlapped. Some tall people must band their waists to put their hands in the box. With these issues in mind, we redesigned our interaction space with transparent plastic glass. As Figure 3.8 shows, this design may not hinder the user's vision. Although there still exists light and shadow issues, because the work space is not properly blocked, the tracking performance overall is quite acceptable.

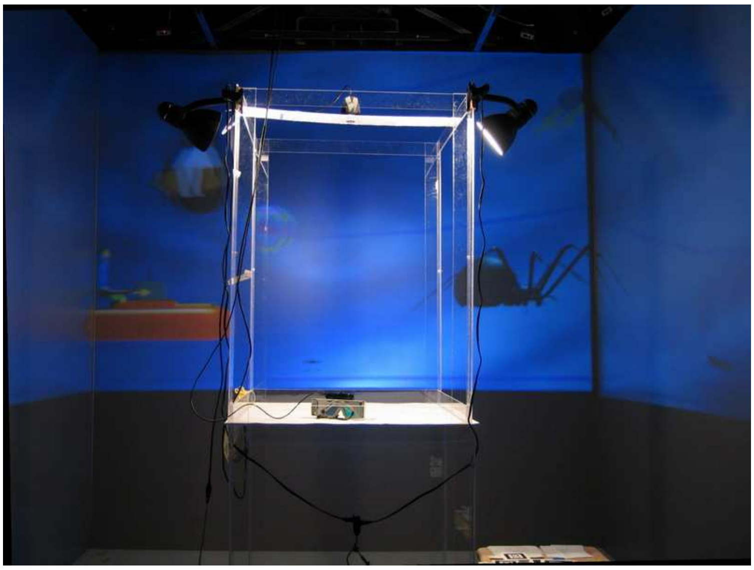


Figure 3.8: *ARDesk*

3.2.6 Configuration

The configuration of the first system is presented in Figure 3.9. The camera is placed over the work desk, three projectors are used for the main view to display a virtual world, and a top projector, when it is available, is used to provide a white rectangle of keeping constant light for the work area. The camera is connected to the AR system, and the tracking information is sent to the VR system over the wireless link. To provide the immersive virtual world, our application uses four DADS systems to control the four screens. The user's hand motion is limited to the space within the work desk. The user can interact with the virtual world using props placed on the work table.

As Figure 3.10 shows, we changed our system configuration a bit for *ARBox*. The top projector was not used any more to brighten the work area. Everything else was identical to the first system configuration.

The system configuration with *ARDesk* is in the same as with *ARBox* because all we did replaced our interaction space with *ARDesk*, which is independent of the system. Details of the experiment with *ARBox* and *ARDesk* are described in Chapter 5.

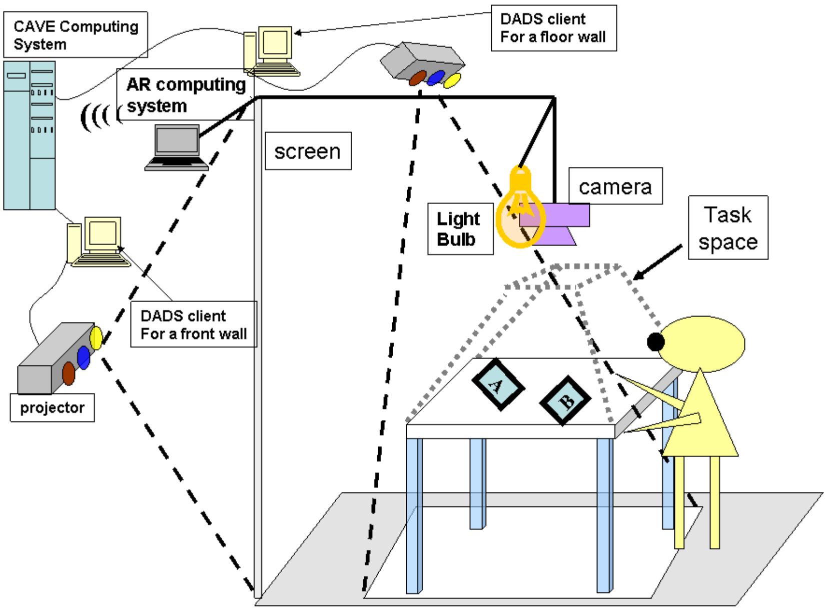


Figure 3.9: The first system configuration

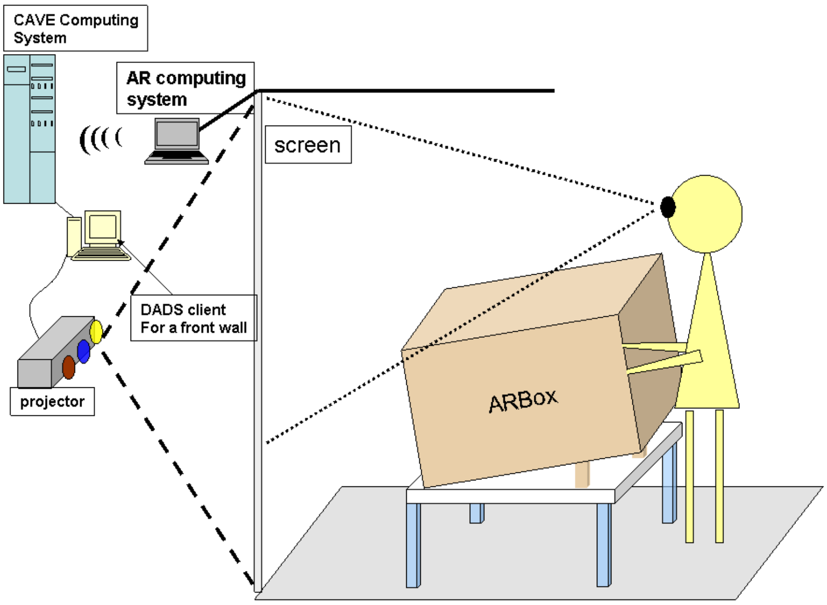


Figure 3.10: The second system configuration

3.3 System Implementation

3.3.1 Procedure

Figure 3.11 shows in overview how our interface system is implemented. Our interface system consists of AR and VR parts, and the VR part includes two subsystems, the *Graphic system* and the *CAVE system*. In this section, we briefly describe our interface procedures, each subsystem being described in detail in following sections.

The ARToolKit-based vision technique tracks a props' position and orientation (i.e., pose) in our interaction space. The camera is calibrated by a pre-defined data file. Pattern markers attached to the props are detected and recognized at each frame by the ARToolKit-based application of the *AR system*. The frame rate is over 30 frames per second. Our application of *AR system* is programmed to map virtual objects loaded into the *Graphic system* with each marker's unique identification(ID). The *AR system* continuously sends the updated pose data of the props to the *Graphic system*, and the *Graphic system* renders properly on the virtual scene virtual images with the pose data. The *CAVE system* projects the updated images onto the screen walls. Finally, users experience the output of their interactions immersively.

3.3.2 Role of Subsystems

- **AR system**

  The *AR system* includes a camera and network communication, as well as an ARToolKit based application. The camera captures every movement on the work area, and the ARToolKit based application uses each video frame captured by the camera to get tracking information. The application sends the analyzed data to the *Graphic system* over the network link. If the user moves her hands with props too fast, the application may not analyze some frames because the patterns on the props are not recognizable from the captured images. In this case, the application sends the pre-

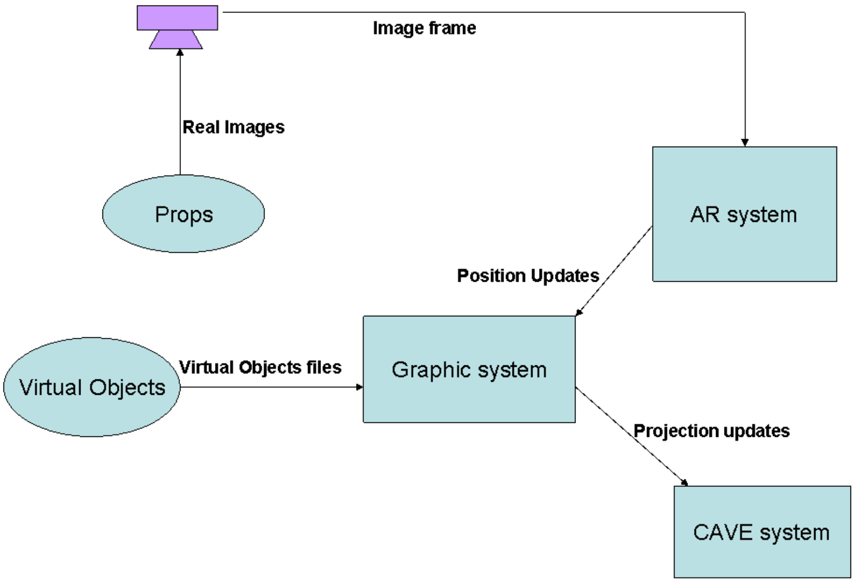


Figure 3.11: Procedure of our interface system

vious tracking information to the *Graphic system*. The actual data structure sent to the *Graphic system* includes the marker's ID, the manipulation mode (such as translation, rotation or scaling), the axis, and the sign information as well as the tracking information.

- **Graphic system**

The *Graphic system* uses a Linux clustering system to control four screen views. The *Graphic system* generates virtual scene and objects from image files. It positions virtual objects on the virtual scene by means of the tracking information sent from the *AR system*. The application on the *Graphic system* is implemented by DIVERSE, mainly the DTK and DPF libraries, and OpenGL Performer^{TM}. DTK is responsible for reading and writing the data structure sent from the *AR system* to *shared memory*. DPF establishes the relationship between the virtual world coordinate system and the real coordinate system with actual tracking data. We use OpenGL Performer^{TM} to generate, with the data processed by DPF, the correct virtual scene and objects.

- **CAVE system**

  In our subsystems, the *CAVE system* generates a physical space, by means of stereoscopic glasses with a head tracker, three 10 x 10 foot projection screens, and our own made interaction space.

  The *CAVE system* displays the final output to the screen walls. Users can see their interaction outputs on three projection screens, and, wearing stereoscopic glasses, they feel immersed in the virtual world. The head tracker attached to the glasses assists in providing the proper viewpoint as the user moves her head this way and that. Every physical movement in our interaction space is captured by the *AR system*, and users feel immersed in the virtual world as if they were directly interacting with virtual objects.

3.4 Summary

We have described in this chapter the core components of the AR and VR parts required to implement our interface system . To facilitate our interface system, we used existing systems and software toolkits, such as the VT-CAVE$^{\mathrm{TM}}$ infrastructure, DADS, DIVERSE and ARToolKit. Basing our work on these components, we proposed new input devices and interaction space for a new 3D interface approach.

Our input devices, i.e. props, are made of hardboard paper with quarter-inch thickness. We attached two handles to the props, making the props very comfortable to handle and light-weight. Making the props card-shaped enhanced visual affordance.

Our Interaction space design was changed twice during the iterative experiment, but the initial system configuration was not changed. Thus, our interaction space is independent from the other systems utilized in the implementation of our interface system.

As shown, implementation of our interface system involves combining two interesting parts, AR and VR. We divided our interface system into three subsystems, the *AR system*, the *Graphic system* and the *CAVE system*, according to their roles. Using existing infrastructure allowed us to concentrate on harmonizing these different subsystems, integrating our interface system into them, and effectively applying our new interaction techniques to the interface system.

Chapter 4

System-specific Interaction Techniques

In immersive VEs, simulator-based interaction techniques depend on the use of controls identical to the controls used in the real world: for example, simulated controls work the same way on simulated vehicles as real controls do on real vehicles. However, interaction techniques for different input devices must be newly developed, in most cases, because these interaction techniques are tightly coupled with the input devices' manipulation methods. In addition, these interaction techniques are developed in various ways even for the same input device. Since our approach is about a new 3D interface system, we need to provide new interaction techniques.

In the following sections, we introduce interaction techniques for fundamental 3D interaction tasks, such as selecting, translating, rotating, and scaling virtual objects, as well as for navigation. We expect that any future practical application based on a CAVE^{TM} system will utilize more promising techniques by iterating the interaction techniques we are proposing. The results from user evaluations presented in Chapter 5 give are encouraging. To begin, we review our interface, describing its design philosophy, and then present the interaction techniques implemented in our interface system.

4.1 Design Philosophy

Unlike typically dedicated VR input devices, which are used solely to facilitate interactions with the computer, the input props used in TUIs operate as a means to give the user *affordance*, defined at Wikipedia (http://www.wikipedia.org) as "a property of an object, or a feature of the immediate environment, that indicates how to interface with that object or feature. The empty space within an open doorway, for instance, affords movement across that threshold. A couch affords the possibility of sitting down on it.". TUI props have two functionalities: they can facilitate interaction with the computer, and they can enable the user to experience abstract spatial relationship concretely [20]. The most common VE input device for two-handed operations is a data glove, using magnetic tracking technology. Data gloves are especially useful in gesture-based interfaces defined entirely by the movements of the user's hands and fingers. However, when using gloves to grab and manipulate virtual objects, users can not experience the tactile feedback that comes from grasping a real-world object. When the user holds a physical tool, haptic feedback provided by the prop itself guides the motion of her hand.

If our interface were designed on a typical TAR system, it might provide a spatially seamless display, such as that in which the output display device is not the projection screen wall separated from physical objects, but a see-through head-worn display. However, if we can invisibly provide affordance to the props on our interface, we can reduce the context switching, which results from the separation display between interaction space and display space.

4.2 Interactions for Virtual Objects

Before we start describing our fundamental 3D interaction techniques, we need to mention two interaction distances, the distance within arm's reach and the distance out of arm's reach. Because our interface is performed with physical objects, requiring that one or two hands directly manipulate virtual objects, it would be hard to handle objects out of arm's reach. If a virtual object is placed within arm's reach, we call it a *near* object; if it is out of arm's reach, we call it a *far* object. Users can manipulate near objects directly and intuitively, just like in the real world, but they cannot manipulate far objects in the real world.

However, in a virtual world, users are able to handle far objects, using range-based interaction techniques developed for VEs. For example, the most popular technique is based on *pointing*, and there are many pointing-based techniques, such as ray-casting, Go-Go, flashlight and fishing-reel techniques [8]. In addition, there are virtual hand based techniques and a world-in-miniature (WIM) technique used interactively to extend the arm's reach. As these early researches showed, there are special 3D interaction techniques to manipulate far objects that can not be manipulated directly. We need to develop more efficient interaction techniques to control even far objects but leave this to future work. In this chapter, we only consider the manipulation of near objects.

4.2.1 Design

Basically, an object has three possible states: selected, grabbed and deselected. For example, the grabbed state is very similar to when an object is being dragged by a mouse in a standard GUI. Typically, users in any VE must have the means to control virtual objects in these three states, just as they use mice on the desktop GUI, clicking buttons or dragging 2D objects. In our approach, since props correspond to virtual objects one to one, users do not need the special interaction techniques required to control all three states of virtual objects in other VR applications. They just grab props, and then, after grabbing them, release them. Since users can easily position props, there can be more possible interaction techniques.

Figure 4.1 illustrates the first interface design. With this design, we used LEGO$^{\text{TM}}$ blocks for the props, and we conducted a pilot study of just the selection and the rotation tasks of

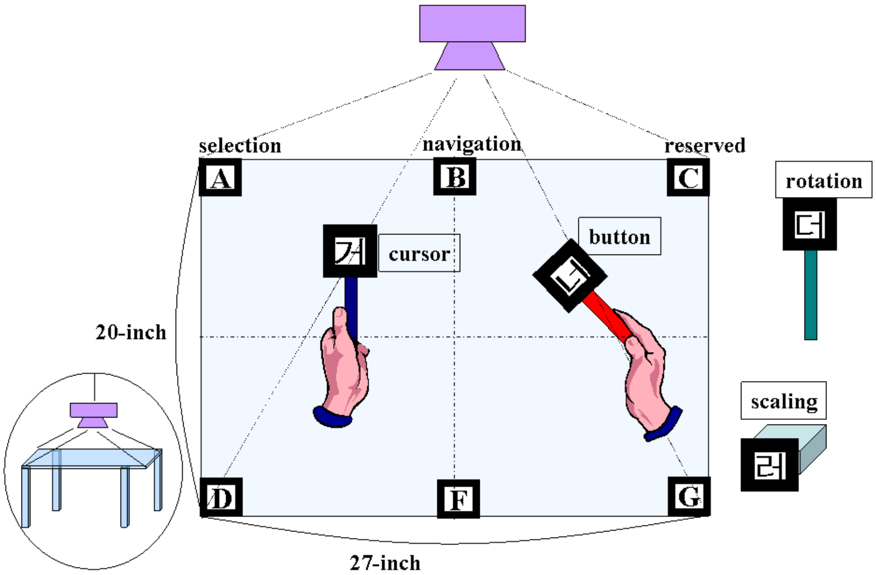


Figure 4.1: The first interface design for selection and rotation

virtual objects.

Our original idea for this design was that markers with **A, B, C, D, F**, and **G** patterns would act as a menu as well as boundaries for the AR work area. If the *button* marker touched the **A** marker, then the current task mode would be changed to *Selection*. If the *cursor* marker points out one of virtual objects and the *button* marker hits the *cursor* marker, then the virtual object that the *cursor* marker is pointing out is selected. Once a virtual object is selected, the user can see just as if the virtual object were attached to the *cursor* marker. Therefore, the virtual object can be moved along with the movement of the prop. If the user wants to place the virtual object in the virtual world, she can hit the *button* marker with the *cursor* marker again. In this case, the task mode would be changed to the default mode in which the user can reselect any virtual object. While grabbing the virtual object, the user can rotate it with the *rotation* marker, i.e., when the task is in the *Selection* mode, the user can use the *rotation* marker instead of the *button* marker.

After we conducted the pilot study for a selection task and one-direction rotation task, we found that our original thinking did not encompass the user's full use of TUI. For example, originally the user can only move one object at a time, and it was also difficult to rotate the object along a specific axis. We decided to come up with another way to improve the interface design from the original thinking.

We still use LEGO^{TM} blocks as *object props*, and we added the paddles as *control props* to control scaling direction as well as rotation direction for virtual objects. Our second idea for the interface design is illustrated in Figure 4.2. Each object prop has two markers, one on its top and another on its bottom. One or the other marker, either top or bottom, is used for virtual object manipulation, and the other marker is used for navigation in the virtual world. Thus, the user does not need to change props when switching between object manipulation and navigation. The control prop, with three markers, can optionally be used to control direction, velocity, distance, the scaling factor, and so forth.

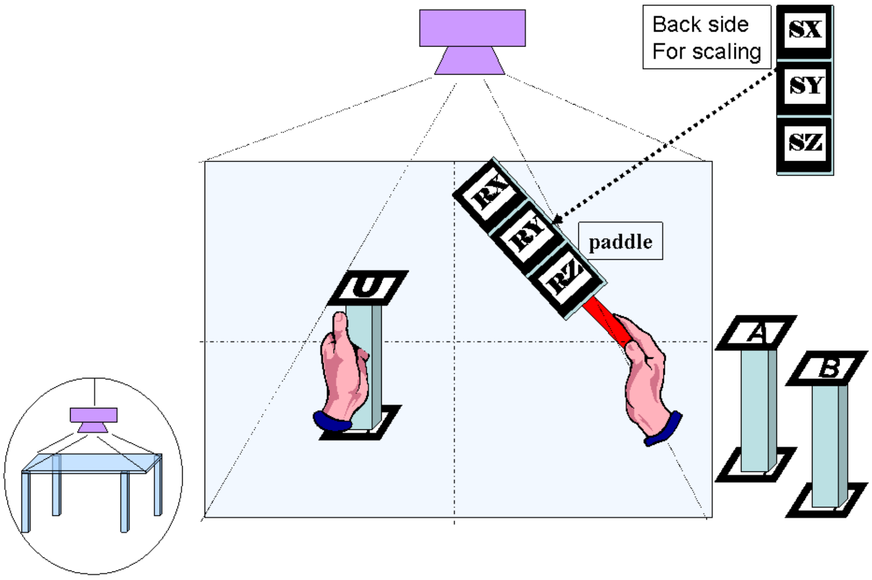


Figure 4.2: Initial Setup for Interactions

In the following section we describe in detail each interaction technique for all of the fundamental tasks .

- **Object Selection**

  The user needs to use only the prop for selecting virtual objects within the near space because each virtual object follows the prop's movement. Figure 4.3 shows how the user moves virtual objects along the path of the corresponding markers. Alternatively, the paddle can act like a mouse button for selection in case the number of virtual objects is too great to have one to one mapping between props and virtual objects. But this paddle technique is not used in the current design.

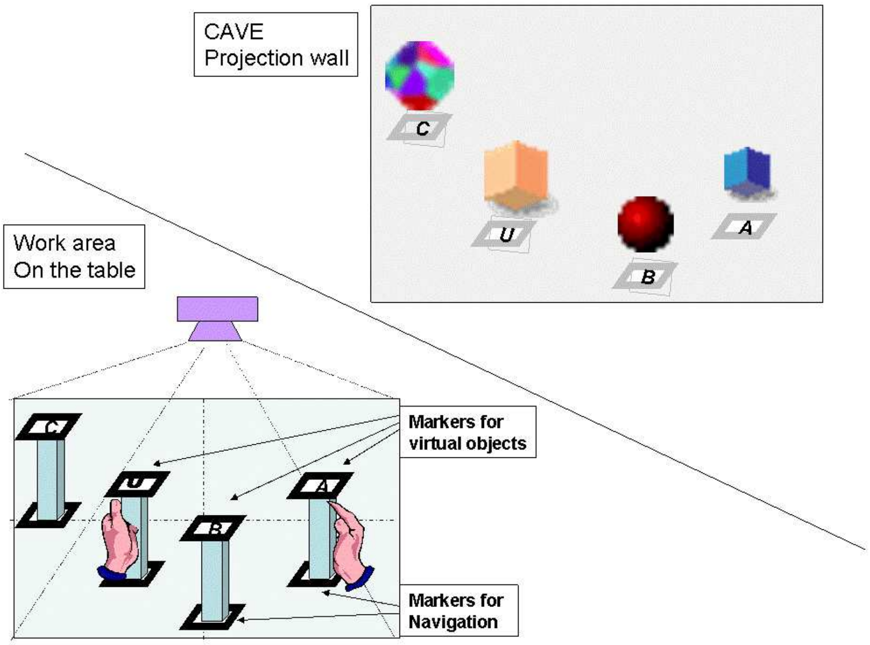


Figure 4.3: Selection

Before we conducted the formative user evaluation, we recognized that the work area was too small to translate virtual objects even in the near space. Because the work area depends on the distance between the camera and the pattern marker on the prop, it occurred to us that if we can widen the distance, we can create a wider work area. Therefore, we changed the design of the object prop, making it the same shape as the control prop, as shown in Figure 4.4.

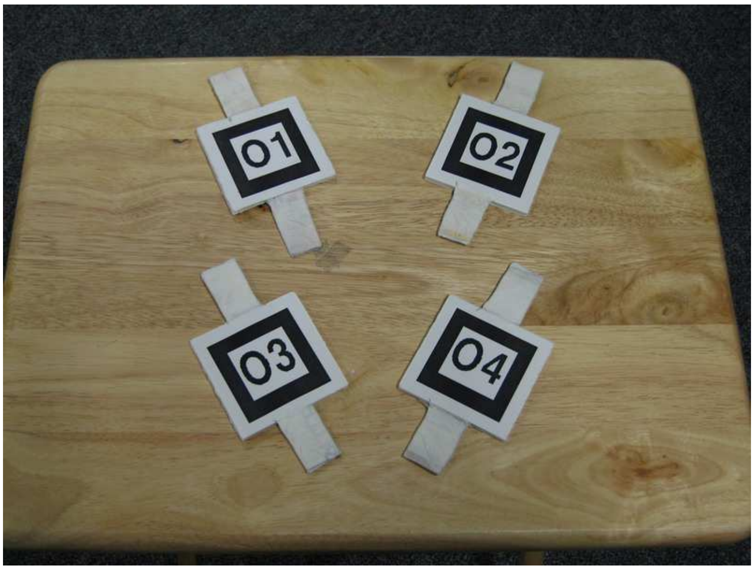


Figure 4.4: New design for object props

- **Object Manipulation**

  Unlike the selection task, the rotation and scaling tasks require another prop for full rotation about the x,y and z axes and precise scaling of objects. For this reason, we designed the paddle prop to control the direction for rotating and scaling virtual objects. As Figure 4.5 shows, one side is designed for the rotation task and the other side is for the scaling task. Three markers present each axis, X, Y and Z. We defined the *detection range* between two props, the object prop and the control prop, so as to facilitate detection of a task mode of translation (default mode), as well as rotation and scaling modes. Details are described in the following paragraphs.

  Rotation As Figure 4.6 shows, each object prop corresponds to a virtual object. When the paddle prop is close to the object prop, the ARToolKit-based application detects that the paddle prop has entered the predefined detection zone. Then the AR application sends a set of information, including a task mode (i.e., a rotation) and an axis about which to rotate, as well as position and orientation data, to the CAVE^{TM}

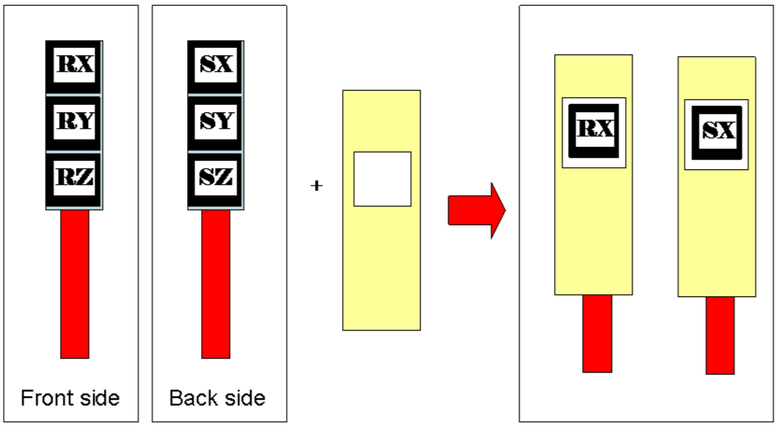


Figure 4.5: Paddle

application. The CAVE^{TM} application uses position and orientation data to rotate the virtual object about the axis sent from the AR application.

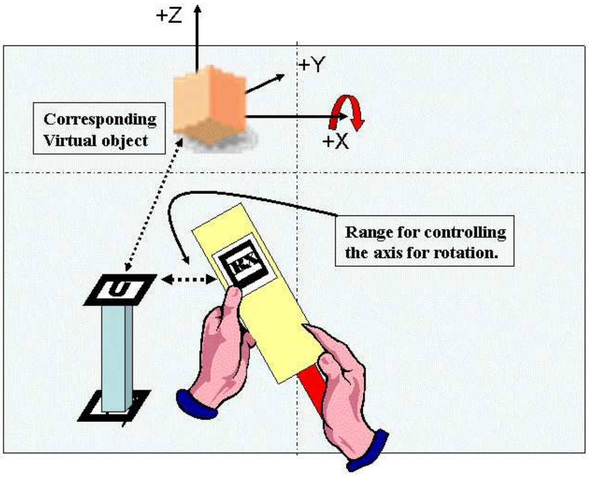

+Z
+Y
+X
Corresponding Virtual object
Range for controlling the axis for rotation.
RX
U

Figure 4.6: Rotation

Scaling As we did for the rotation task, we use the paddle to indicate the scale direction for virtual objects. As Figure 4.7 shows, the interaction technique for scaling is performed in the same way as the rotation task. After the AR application detects the paddle for scaling, the AR application sends information concerning the rotation task and scaling task mode, instead of the rotation task mode. The CAVE^{TM} application resizes the corresponding virtual object according to the axis information sent from the AR application. The AR application recognizes the position of the paddle when it is properly positioned close to the object prop. The first position can be the origin for scaling. As the paddle get closer and closer to the object prop, the virtual object gets smaller toward the indicated axis. If the paddle moves in the opposite direction, the virtual object gets bigger. These operations must be performed within the detection range between the object prop and the control prop; otherwise, the task mode is changed to the translation mode (default mode), and the user cannot perform either the rotation task or the scaling task.

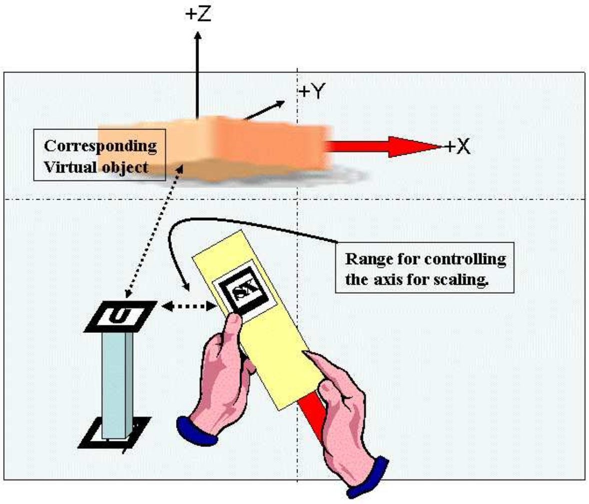

+Z
+Y
+X
Corresponding Virtual object
Range for controlling the axis for scaling.

Figure 4.7: Scaling

We developed the paddle for rotating and scaling virtual objects, but with this paddle the user can only rotate or scale the object in one direction. As the research [32] shows, to resize a virtual object we must consider all cases, such as along only one axis, along two composite axes, or along all three axes. The same applies to the scaling technique. Basing our decision on these findings, we redesigned the paddle prop, as shown in Figure 4.8.

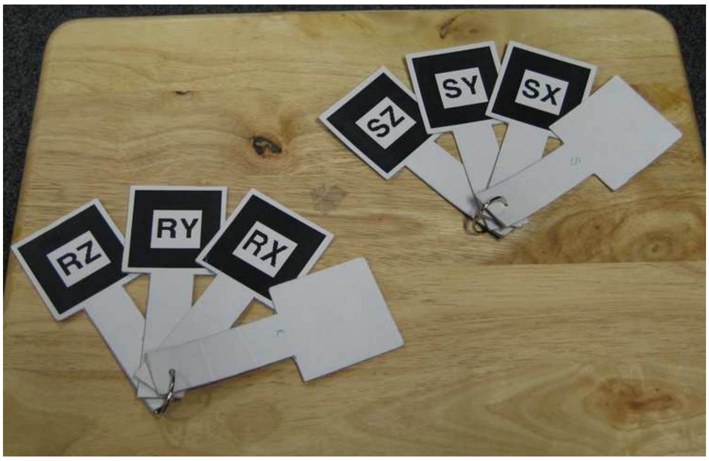


Figure 4.8: New designed paddle for rotation and scaling

4.3 Navigation for Virtual Space

Navigation is required to search for information on web sites or in documents, as well as in traveling through virtual space. When dealing with actual navigation tasks, the size of various virtual spaces usually exceeds the physical interface, and the user feels a need to navigate these virtual spaces. Sometimes the user moves herself in the space (*Egocentric Travel*), but in other cases the world moves around the user (*Exocentric Travel*). Thus, in the CAVE^{TM} room, the user either physically moves around the room, or stays in place, using her hands and/or her head to do the navigation. Either way, the user needs two types of functions to perform the navigation tasks; in order to switch the navigation planes, or start and stop movement, discrete functionality is required; in order to keep moving and rotate the virtual scene, continuous functionality is needed.

When the user walks about in the CAVE^{TM} room, her movement controls these functionalities; but when she stays in one place, she must use input devices connected to the CAVE^{TM} system to control them. Generally, a wand, equipped with a few buttons (for discrete events) and a joystick (for continuous events), is used for this purpose. As a tracking module is integrated into the wand, every tracked position and orientation data can be delivered to the CAVE^{TM} system, which renders the virtual scene in accordance with the data. With regards to the current interface, since our interaction technique for navigation is based on the hand operation, we will only consider the *E*xocentric Travel task, which is either moving the world or scaling the world.

4.3.1 Design

Figure 4.9 shows the working space with the origin in the middle of the work area. As we mentioned before, the original prop for navigation was an object prop with two markers, one on the top and one on the bottom. The markers top and bottom were used for a virtual object or for navigation, respectively. We finally changed the prop for navigation to the same shape as the object prop, as shown in Figure 4.10. When the user puts the prop on the origin, the virtual space does not move. As the user moves the prop away from the origin, the virtual scene starts moving along with prop's movement. Navigation speed is controlled by the distance between the prop and the origin. For example, as the prop moves closer to the origin, the navigation speed is reduced. This navigation technique is borrowed from joystick technology. In our interface design, the user is not allowed to walk around the physical place. Because both the camera's position and the work desk are fixed, the camera cannot detect the user's hand movements if the user moves about the room.

4.4 System Control Technique

At first we considered only fundamental 3D interaction tasks, such as object selection and manipulation, and navigation. However, we realized in time that we needed a menu system to change from the current task to the next task or to reset the virtual scene for user evaluations. In the traditional test bed for VEs, an investigator usually changes the current state to another one in person, at the computer, because of the lack of technical functionalities of input trackers, such as wands with just four buttons and the typical joystick. Otherwise, the investigator employs a 2D style menu system. But we can easily give any functionality to

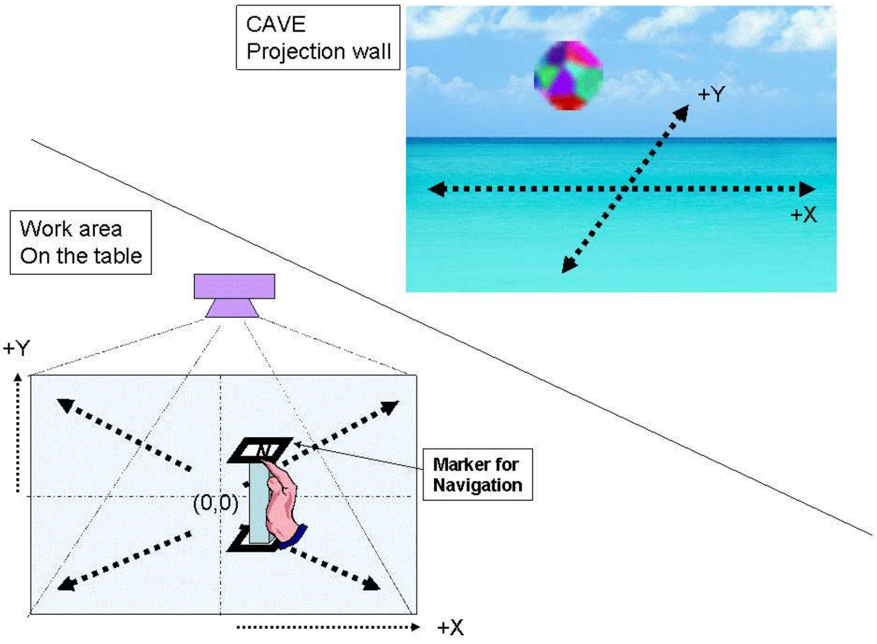


Figure 4.9: Navigation

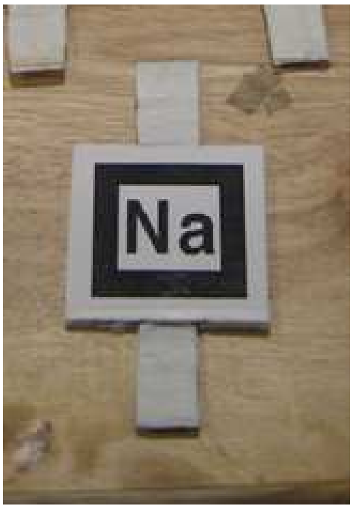


Figure 4.10: Redesigned prop for navigation

a marker and make as many markers as we need for our application. Even though we did not provide the user with the ability to control the system herself, because we estimated that it was unnecessary for this study, this kind system control can be applied in other applications.

4.4.1 Design

Since we needed just a few functionalities for our user evaluations, we made do with same shape object and navigation props, as shown in Figure 4.11. We could even use a cube, instead of the paddle, for convenience.

For controlling the system, we just added to these markers' unique IDs and let the AR application map each ID to the specific functionality, such as to reset the scene, to start each task (i.e., Task 1, Task 2, Task 3 and Task 4, see Chapter 5), and to remove all objects from the virtual scene. If we give proper patterns (i.e., visual affordance suitable for system control) to each marker, users can control the system by themselves.

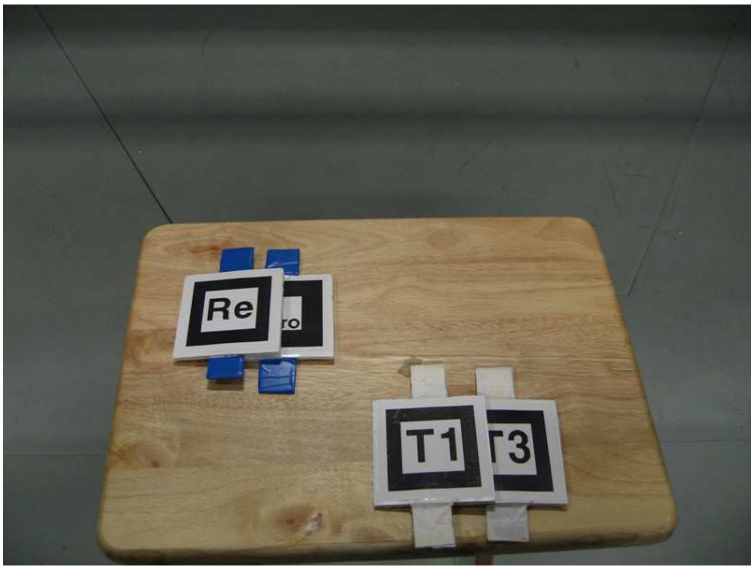


Figure 4.11: Props for system control

4.5 Discussion

An interesting characteristic of this interface is that it can generate, to some extent, as many interaction methods as a designer wants. For example, we can use different size paddles (or props), each size giving a sort of haptic memory to the user.

We presented a few interaction techniques devised simply but effectively, so that 3DUI designers can easily work on developing better interaction techniques. In the design phase, we still have some issues that need to be addressed.

The interaction techniques must depend on the input devices. We tried to bring features of tangible interfaces, such as affordances and two-handed operations, including intuitiveness of natural interfaces, into the projection-based VR system. For future work, we may use other shapes of physical objects, which can give physical affordances to the user as well as visual affordances. We may apply sound effect directly to our input props so that users can easily recognize what they are currently grabbing without actually watching the physical objects. We believe that our approach is extensible toward robust 3D UI.

In the following sections, we discuss the strengths and the weaknesses of the current design with regards to input tools and interaction techniques unique to our interface design, and give some suggestions for future design.

4.5.1 The Strengths

The strengths of the current interface design are straightforward, including all of the strengths of TUIs in general, which we mentioned in Chapter 2. These strengths include the user's ability to take advantage of the immersive experience made available in the CAVE$^{\text{TM}}$ system.

- **One-to-One mapping with a virtual object**

  Since all of props are made of hardboard paper, they are light weight and easy to handle. The user does not become fatigued with these props, unlike conventional electromagnetic devices, even after manipulating them for a long time. Functionality is more natural than with other designs. But we can still manipulate virtual objects in ways other than one-to-one mapping.

- **Handling multiple objects simultaneously**

  We added two handles to our prop design to enable users simultaneously to handle multiple objects, as many props as a user can grab at once. Even when several props are held in the same hand at once, they are not overlapped easily, and fingers rarely occlude the patters on the props.

- **Easily regenerated with low-cost**

  Since our input tools consist of patterns printed-out on paper and card-shaped hardboard paper, they can be easily and simply regenerated when worn out. In addition, whenever the patterns are not suitable for a particular user, the designer can simply redraw them. This is not the case with wands and data gloves, common electromagnetic input devices in immersive VEs, which cannot be redesigned and are expensive to replace.

- **Freedom of placement**

  Our props are wireless and cable free and can be placed anywhere in the CAVE$^{\mathrm{TM}}$ room, even hung from the ceiling or attached to the body. They can be placed intentionally underfoot, for purpose of occlusion. They can be dropped on the floor when not in use.

4.5.2 The Weaknesses

- **Unreliable video-based tracking**

  There are some limitations in extending its interaction methods because our interface uses a marker-based AR technique sole to get tracking information. For example, the video-based tracking technique can be unreliable, compared to electromagnetic or infrared-vision based trackers, due often to wrong detection because of shadows and non-detection because of insufficient light. In addition, unintentional occlusion can hinder the user's reactions.

- **Unsuitable for complicated interactions**

  We had to devise more delicate interaction techniques for rotation and scaling tasks. For example, we had to work out the relationship of the distance between two props, the *object prop* and the *control prop*, to control scaling up or down. This relationship must be made consistent so that the user understands precisely how to scale up and down the virtual object. Our design, based on the dynamic positions of two props on the work area, sometimes made users confused.

  We ran into the same kind of problem with navigation. The distance between the origin, in the middle of the work area, and the prop is used to determine the traveling speed. Since the response time for displaying the output of interactions was a little slow, users experienced the actual navigation speed one step too late. At first, the user feels it is too late, and moves the navigation prop away from the origin. However, the navigation speed is actually fast enough; the user simply experienced it too late. Then the user feels it is too fast, and moves the navigation prop close to the origin again. Thus, the current interaction techniques are not quite suitable for more complicated VR tasks. However, if we were to indicate the fixed position on the work area and modify the interaction techniques accordingly, this weakness might be easily solved.

- **Force feedback**

  The current interface does not provide any force feedback but the graspable property because the current prop is only visually mapped to the virtual object. In order to support the force feedback, extra electronic sensors are required as a way. We will consider applying electronic sensors to our input tools in future work.

- **Context-switching**

This weakness is natural for our interface because two spaces, interaction space and display space, are separated. However, cognitive load, as this affects context-switching, can be reduced. It all depends on how one designs the physical environment (i.e., the interaction space). For example, we designed *ARBox* to prevent users from seeing their hands and interface tools (i.e., the props); they can watch only the virtual scene on the screen wall. However, the user study showed that it would be better to allow the user to see the physical objects to some extent, even though context-switching may be the result.

Chapter 5

User Study

We performed iterative user studies in three phases: the pilot study with the initial approach, the formative user evaluation with *ARBox*, and the summative user evaluation with *ARDesk*. Our purpose was to provide a rapid and early evaluation of our utilization of the 3DUI in the CAVE$^{\text{TM}}$ system. We assumed that our interface might reduce the immersive experience, due to separation of interaction space and display space, and anticipated receiving feedback regarding interface design, experiment design, and interaction techniques. We conducted the user studies, focusing on the effects of our interface design itself, without comparing it with other interface based interaction techniques, because our interaction techniques are developed on the new interface, and these initial interaction techniques can be improved in future. Therefore, basing our future research on the results from the three experiments, we will improve our interface interaction techniques and compare them with wand-based and data gloves-based interaction techniques. Through this iterative process we hope to make our interface system more robust and our interaction techniques more effective and efficient.

5.1 Pilot Study

The two goals of our pilot study were to examine possible interaction techniques and to get user feedback about the initial interface system. As a result of having done the pilot study, we feel that our approach is effective and already a contribution to the field of 3D user interfaces.

Since our interface is new, we needed to attempt the pilot study to know how people using our interface in the semi-immersive VE are made to feel. The pilot study was performed using our our first interface system design, shown in Figure 3.9. For a software test bed we designed a virtual scene with a couple of virtual objects and a simple background with a sky and a plane. Four subjects participated in this pilot study. The primary focus was to observe their performances, movements, and preferences. Half of the subjects already had some experiences using a wand in the CAVE^{TM} system. The other subjects had used various applications in the computer vision area, but not in VEs. Before the actual experiment, four subjects were asked to experience the wand-based interface using a simple *ray-casting* interaction technique for object selection and manipulation. They were also asked to try the navigation task using a joystick on a wand. Since we wondered if our interface system would work for object manipulation as a 3D interface in the CAVE^{TM} system, we asked users first to perform the wand-based interface, which had already been evaluated in other earlier studies, and then our interface. Even though the pilot study utilized two different interfaces, it focused on the effectiveness of our interface, not the comparison of the two interfaces.

5.1.1 Procedure

Participants were first asked to fill out a *pre*-questionnaire to determine their backgrounds, such as age, vision(eyesight), and 3D experiences. The evaluator gave them some instructions before they performed the first task. They were allowed to talk out loud and freely talk to the evaluator about how they felt during the experiment. The evaluator recorded their every comment by taking notes. After the experiment, they were asked to fill out a *post*-questionnaire designed to measure their personal preferences. The results showed that all of them preferred our interface to the wand-based interface for the following reasons:

- The graspable tools are very intuitive, especially for the selection task, and no training is required.
- The props are lighter and easier to handle than the wand.

They argued, however, that the interface was somewhat disconcerting. For example, when they manipulated more than two objects simultaneously, overall movements of virtual objects slowed down. Sometimes, virtual objects did not move, despite the fact that the users were moving the props. We observed that the camera's quality and performance significantly affected AR system's ability to recognize markers within the frame. Since the extra light bulb had been positioned above the middle of the work area, there existed shadows generated by the props as well as by the user's hands. These shadows caused faulty detection by the AR system . The marker's information was sometimes lost because its movement was too fast for the camera to capture. In addition, the frame was often frozen due to the short disconnection of wireless networking, thereby increasing the overall latency. We describe details of user feedback in the following section.

5.1.2 User Feedback

Subjects were interviewed in person after completing the post-questionnaire (See Appendix A). Overall user feedback showed that they felt more intuitive when they could physically handle props to perform their virtual tasks and explore the virtual world. However, they indicated some problems with our initial system design.

- **Position of the extra light**

The position of the extra light obstructed the users' lines of sight and reduced their immersion. It often caused unexpected shadows to affect marker detection. One solution would be to use four lights located in each corner of the CAVE$^{\mathrm{TM}}$ walls to remove shadows. However, these extra lights definitely reduce the user's immersion in the virtual environment if she still watches the lights while doing her tasks. Another solution would be to design a self-contained light prop, to which markers can be attached. Since ARToolKit is very sensitive to the marker's material, we may need to test various lights and materials. We assign this to future work.

- **Limitation of the interaction area**

  The work area was too small for some tasks, especially navigation, resulting in tiredness. Occasionally, subjects did not realize that their hands were out of the work area, and thereby out of the camera's view.

- **Use of props with irrelevant patterns**

  Subjects liked handling props for selection, rotation and scaling tasks. However, they had a hard time remembering which prop was assigned to which virtual object, especially when they needed to manipulate a lot of virtual objects. Because the initial patterns were very simple they failed to provide sufficient visual affordance to the users.

Besides these, there were some issues regarding the implementation of interaction techniques. For example, users had to move the prop to the origin in order to stop navigation, and, unlike with a joystick on a wand, with our props there is no force feedback to indicate whether or not they had reached the origin.

5.2 Formative Evaluation

Although our test bed and system design were very simple for the pilot study, participants provided positive feedbacks; for example, they thought that our interface was quite intuitive and the manipulative techniques interesting. However, besides the light and shadow issues, we also observed that our initial prop design was not quite comfortable enough for users confined within a limited work space. As a result of on observations made during the pilot study, we redesigned our experiment, and made the *ARBox*. We also provided a new post-questionnaire to get objective measurements as well as subjective measurements. We conducted the formative user evaluation of the *ARBox*, our aim being to alleviate the current issues.

5.2.1 Participants

The total of 10 students from various backgrounds participated as subjects. Some of them already had experience with stereoscopic displays and 3D interfaces, but the others were novices in VEs.

5.2.2 Environment and Equipment

Display and tracking devices The VT-CAVE^{TM} system (Figure 5.1) was used in this study for the semi-immersive virtual environment. It consists of 10x10 feet projection screens, *Electrohome Marquis 8000* projectors, *Ascension* trackers with tethered electromagnetic sensors, and *Stereographics*' LCD stereo shutter glasses used to display the stereoscopic images going to the eye. An *Intersense IS900* tracking system supports the head tracking information (Figure 5.2).

Props As we described in Chapter 3, the props used for this user evaluation are made of hardboard paper with printed-out patterns used according to props' objectives. Figure 5.3 shows the props for object manipulation and navigation tasks, and Figure 5.4 shows the props for the system control task.

As Figure 5.5 and Figure 5.6 for object manipulation, i.e. rotation and scaling, show, the

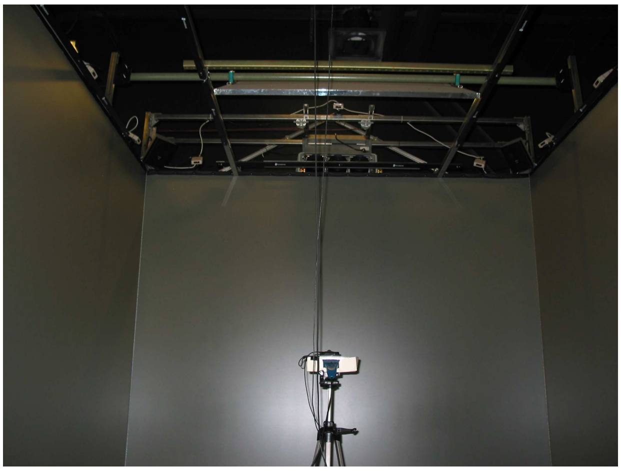


Figure 5.1: VT-CAVE^{TM} projection walls

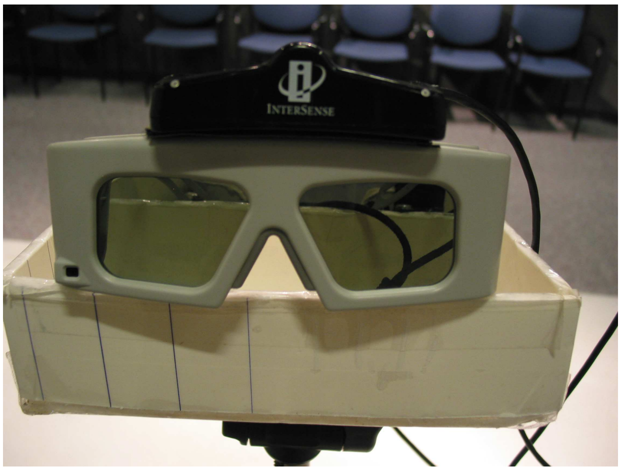


Figure 5.2: Head tracker and shutter glasses

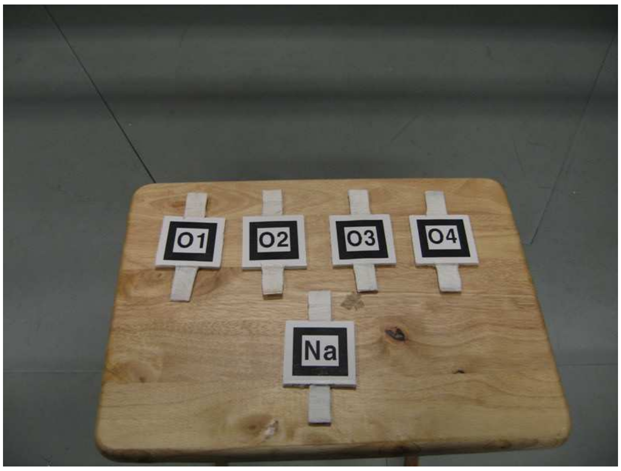


Figure 5.3: Object and navigation props

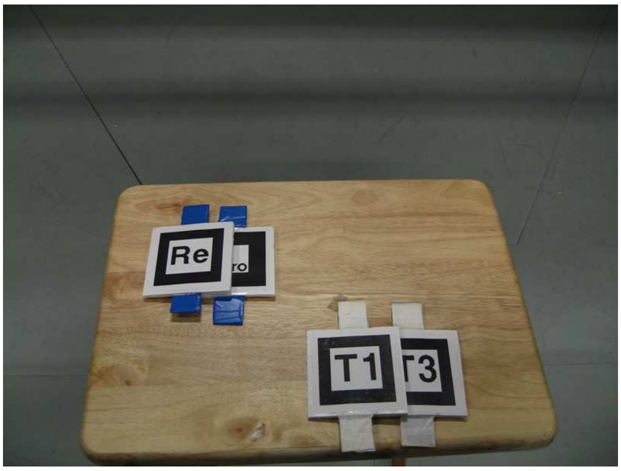


Figure 5.4: Control Props

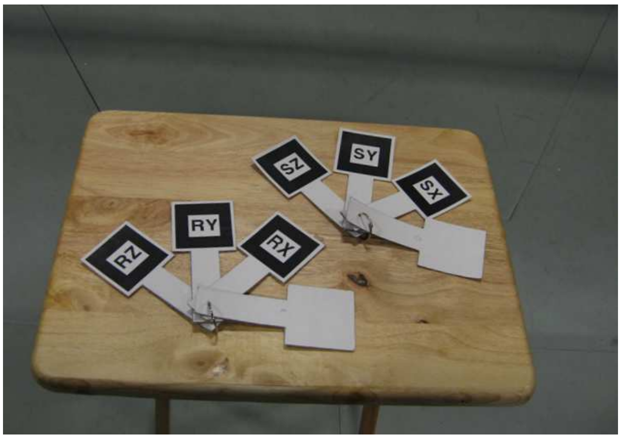


Figure 5.5: First props for controlling

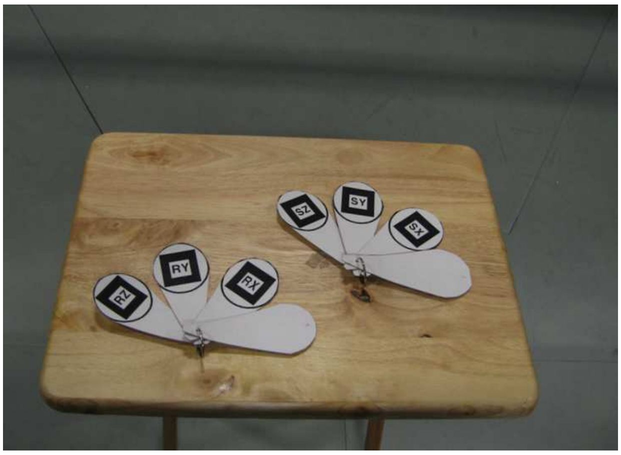


Figure 5.6: Second props for controlling

props are card-shaped, which allows them to be easily switched from one and another marker, representing one axis (i.e., X, Y and Z). The user can easily indicate one or more axes along which she wants to rotate or scale the virtual object.

ARBox In our interface system, besides the CAVE$^{\text{TM}}$ system and stereoscopic glasses, all necessary devices are placed in the *ARBox*. As Figure 5.7 shows, we used the big packing box 29.5"x27"x24" for the main interaction space. We also put a small box 11"x9"x6" on it to widen the view frame because the longer the distance between the camera and the props, the wider the view frame, thereby getting the wider interaction space. We used a paper cup to fix the camera on the small box, and, to solve the light issue, we put two bulbs underneath the top of the box. Since five sides are blocked from the outside, the user's hand movement must be constrained within the box.

5.2.3 Procedure

The formative user evaluation consisted of two sessions, the training session and the actual task session. We designed four tasks for each session. After participants filled out the pre-questionnaire, they participated in these two sessions. During the training session,

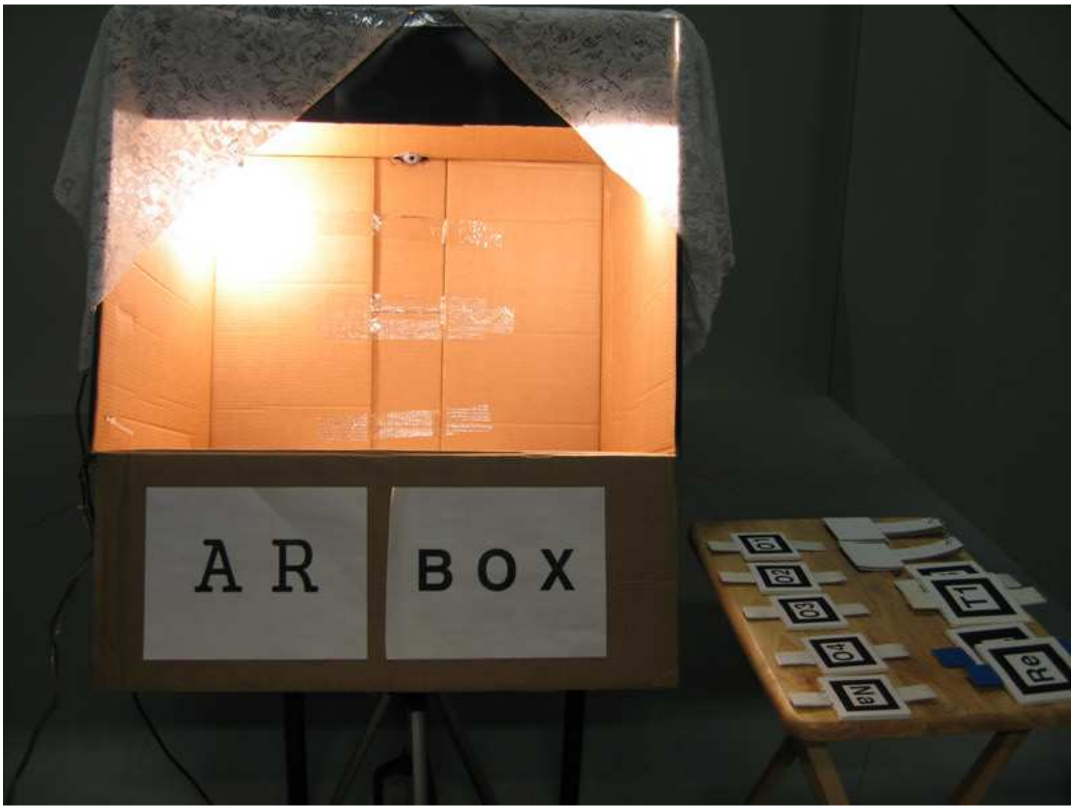


Figure 5.7: ARBox

participants could practice using our interface as much as they wanted. During each experiment session, they were asked to complete four tasks, and the evaluator recorded the results, whether success or failure for each task, including the taken time to complete each task. After the participants finished two sessions, they were asked to fill out the post-questionnaire and have an interview with the evaluator. They provide comments or suggestions based on their experiences. The feedback of individual participants is the most important information gleaned from this experiment because we are still working on improvements to our interface design based on these results. At the end of the experiment, the participants evaluated our interface system with personal ratings for each question, according to their preferences and efforts in the experiment.

- **Software test bed**

  We newly developed a software test bed to evaluate our interface, with actual scenarios for each task. To control the software test bed program, menu props were used. They allowed the evaluator to reset the virtual scene and change the current task to the next one. Therefore, the evaluator did not need to go to the computer to control the system during the experiment. The evaluator measured the experiment time and marked each subject's performance as either "pass" or "fail".

1. **Training session**

   The scene for the training session is illustrated in Figure 5.8. As the figure shows, four objects are manipulated by four object props. During this session, the user experienced four interaction techniques, i.e., selection/translation, rotation, scaling and navigation. Participants can be trained until they are comfortable with our interface system and familiar with handling props.

2. **Task session**

   Task 1 *Task 1* includes object selection and translation tasks. As Figure 5.10 shows,

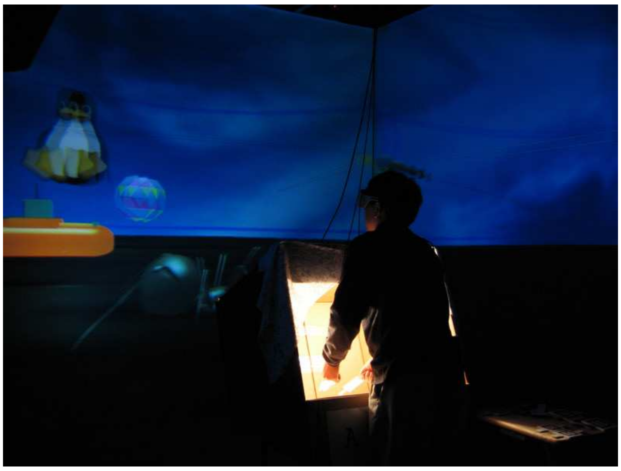


Figure 5.8: The scene for training session

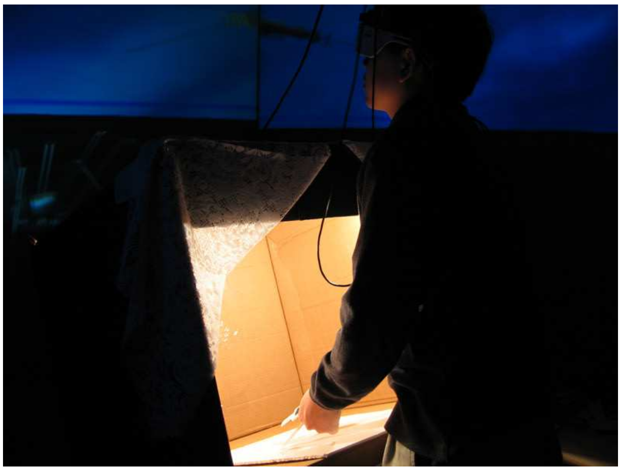


Figure 5.9: A user with props within *ARBox*

one ball object comes out next to the stool object in the virtual world when the evaluator starts with showing the menu prop to the camera for *Task 1*. To complete *Task 1*, the user should put the ball on the stool. When the user succeeds in *Task 1*, the scene is displayed as Figure 5.11. If the subject can not make this happen due to fatigue or other reasons, she asks the evaluator either to restart the current task or discontinue it. If she would like to restart the current work, the evaluator can reset the scene for *Task 1* with the *reset prop*.

Task 2 *Task 2* includes the rotation task, requiring that the participant rotate a virtual object and say three hidden words, which can be revealed by rotating the virtual object. As Figure 5.12 shows, the object has three axes, and one word exists at the end of each axis. The evaluator starts measuring the time when the user starts rotating the virtual object, and ends measuring the time when the user says three words correctly.

Task 3 *Task 3* includes the scaling task, in which the participant scales up or down a virtual object along an axis or along composite axes. Three small yellow cubes are positioned a short distance from the end of the object in the middle of the scene. *Task*

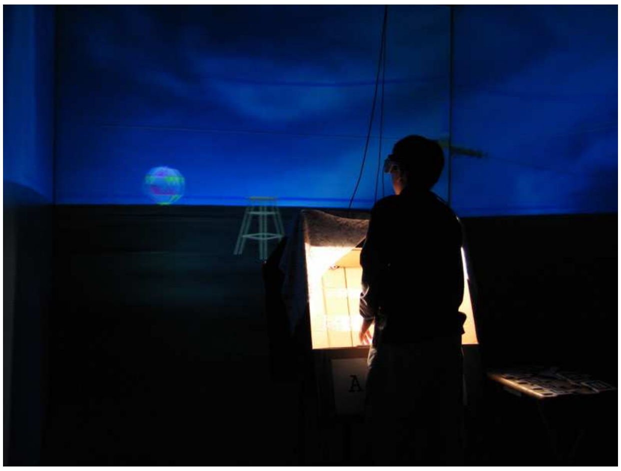


Figure 5.10: The first scene for *Task1*

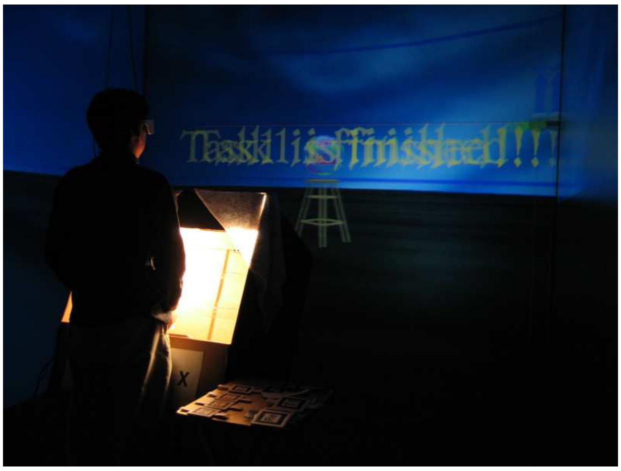


Figure 5.11: After *Task1* is succeeded

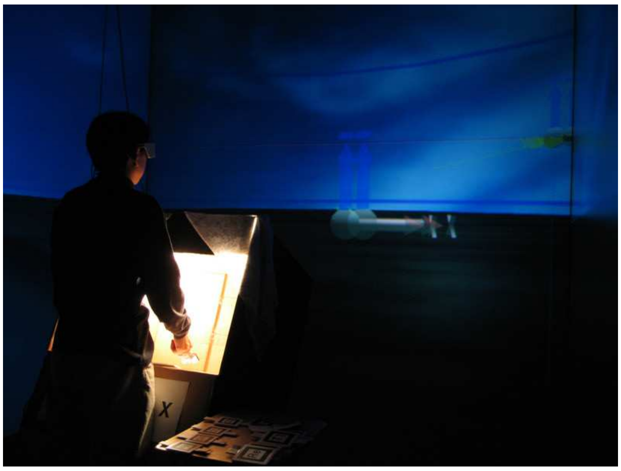


Figure 5.12: The first scene for *Task2*

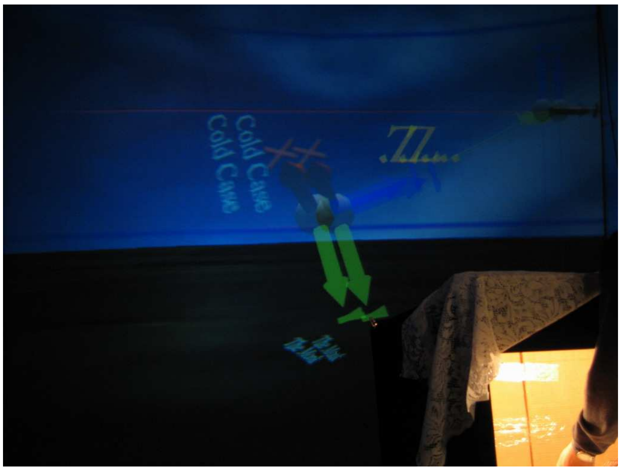


Figure 5.13: After *Task 2* is succeeded

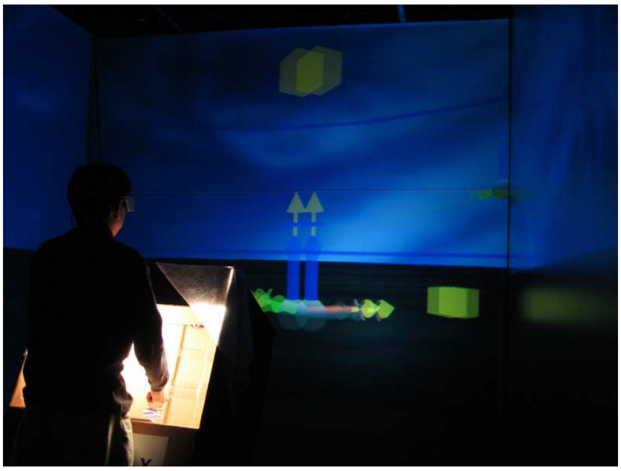


Figure 5.14: The first scene for *Task 3*

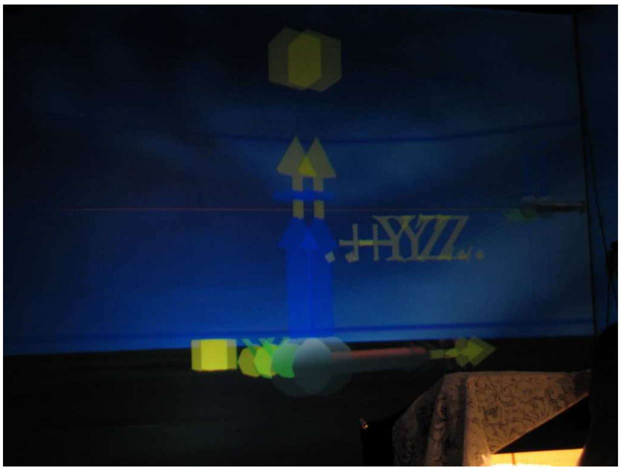


Figure 5.15: After one cube of *X axis* is removed

3 is finished when the user removes three yellow cubes from the scene only by scaling the virtual object. Each cube is removed from the scene when the arrow of each axis touches the cube. The user manipulates the *scaling* prop to stretch the arrow to the specific axis. The scaling factor is fixed in this study because our aim was to observe whether or not the user could complete the task only by handling the *scaling* prop.

Task 4 *Task 4* includes the navigation task, in which the participant navigates the virtual space by hand. We designed the virtual scene very simply for the navigation task. When *Task 4* starts, one virtual object and one sentence are displayed so far from the user's position that she cannot recognize the object nor read the sentence at all from the current viewpoint. The user must pull the background closer and closer her position until she can read the sentence and recognize what the object is. When the user reads the sentence, says what the object is, and stops moving the background, then *Task 4* is completed.

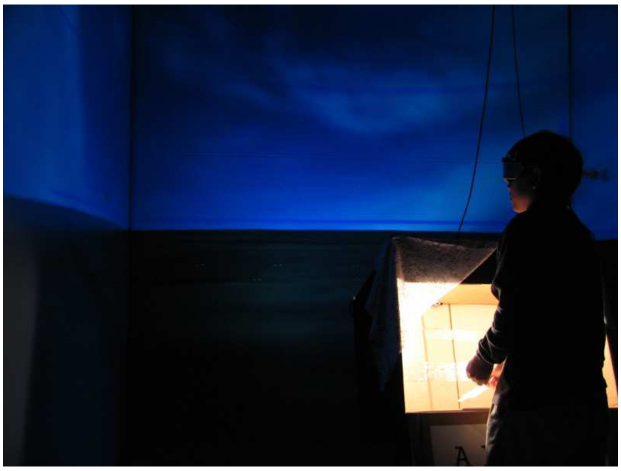


Figure 5.16: The first scene for *Task 4*

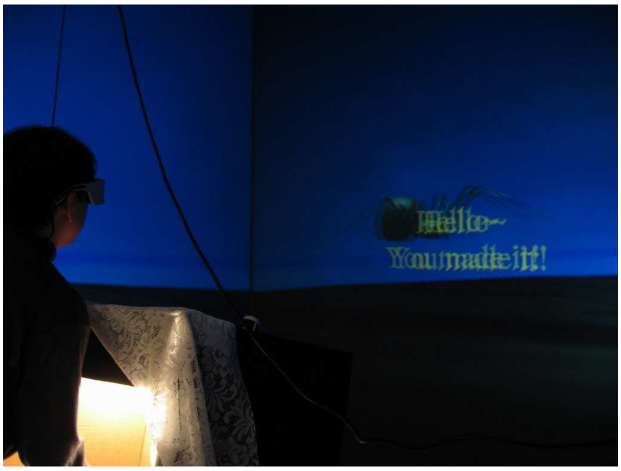


Figure 5.17: After *Task 4* is succeeded

5.2.4 Results

Most subjects preferred the interaction techniques designed for the translation and the navigation task to the rotation and the scaling task. The preference rating scale in the post-questionnaire is from 1 to 7, the larger the number the more positive the reaction. Most of participants said that the interaction techniques for the rotation and the scaling tasks were not intuitive enough and the props harder to manipulate than they had expected, and then we had intended. As Figure 5.18 shows, the participants' personal preference was highest in the control and speed to completion categories (see Appendix A for post-questionnaire). Overall tiredness was also higher in the rotation and the scaling task than in the translation and the navigation task (as Figure 5.19 shows, the tiredness is rated from 1 to 7, with 7 being the most tired).

During the course of this study, we found that our interface was preferred by novices to the CAVE^{TM} environment, who typically were more enthusiastic about the project than experienced users((Figure 5.20).

Figure 5.21 shows that the total time taken for this study was shorter in novices than in non-novices. Though we did not consider their backgrounds when we recruited participants,

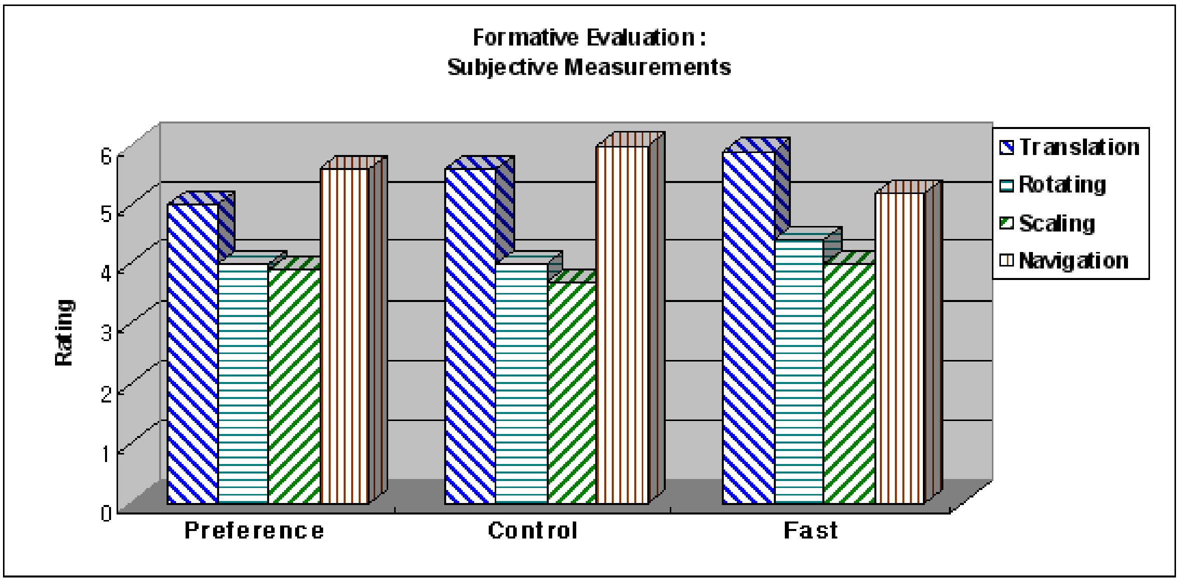


Figure 5.18: Formative User Evaluation: Subjective Ratings

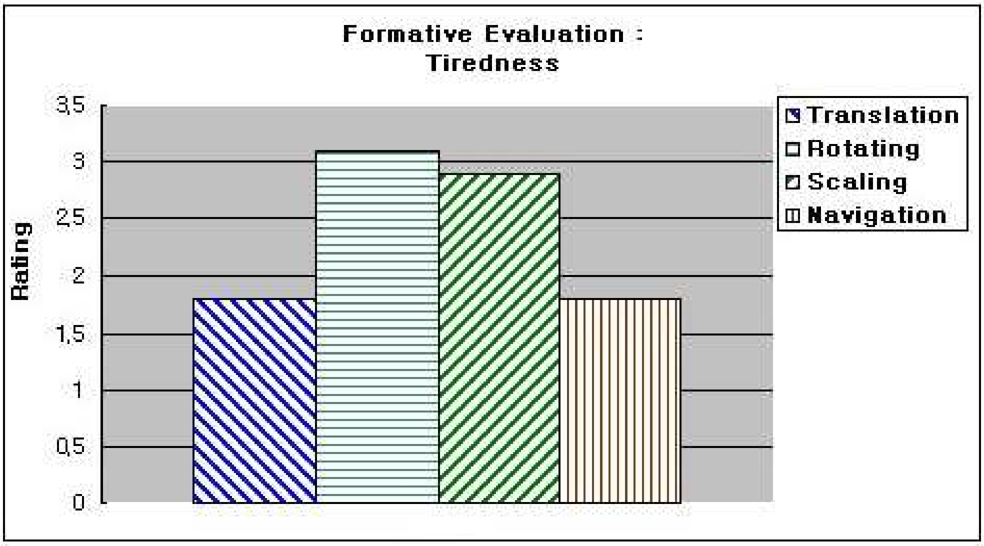


Figure 5.19: Formative User Evaluation: Overall Tiredness

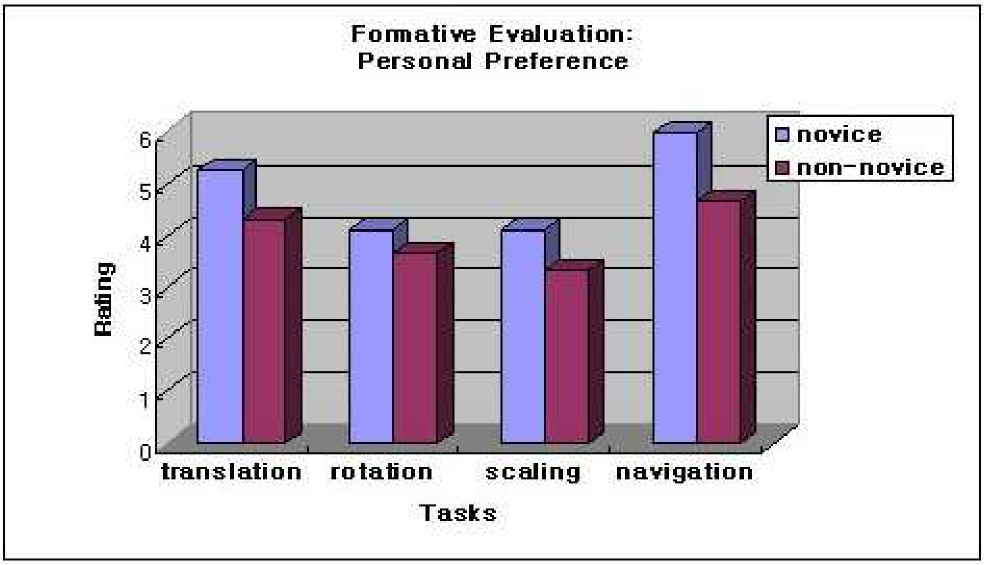


Figure 5.20: Formative User Evaluation: Preference in Novice vs. Non-novice

non-novice subjects comprised three of the ten participants. Although this comparison was conducted with three novices and seven non-novices, we assume that novices were simply more impressed by a new tangible user interface in the CAVE^{TM} system than were non-novices.

As we mentioned before, subjects were asked to train until they felt familiar with our interface system. Non-novices sometimes spent more time than novices in the training session because they tried to compare our interface with other interfaces, with which they had experienced before. However, Figure 5.22 shows that novices finished faster than non-novice in both sessions.

Thus, novices of 3D interfaces in VEs learned our interface rather quickly. We also observed that the experienced users had some trouble with 3D interfaces different from the ones they were used to, and that they spent a lot of time comparing our interface with other interfaces. Both novice and non-novice users said that they were uncertain whether or not they were doing well when their interactions were displayed unexpectedly, because they could not see their hands and their physical manipulations. The *ARBox* was basically designed to prevent

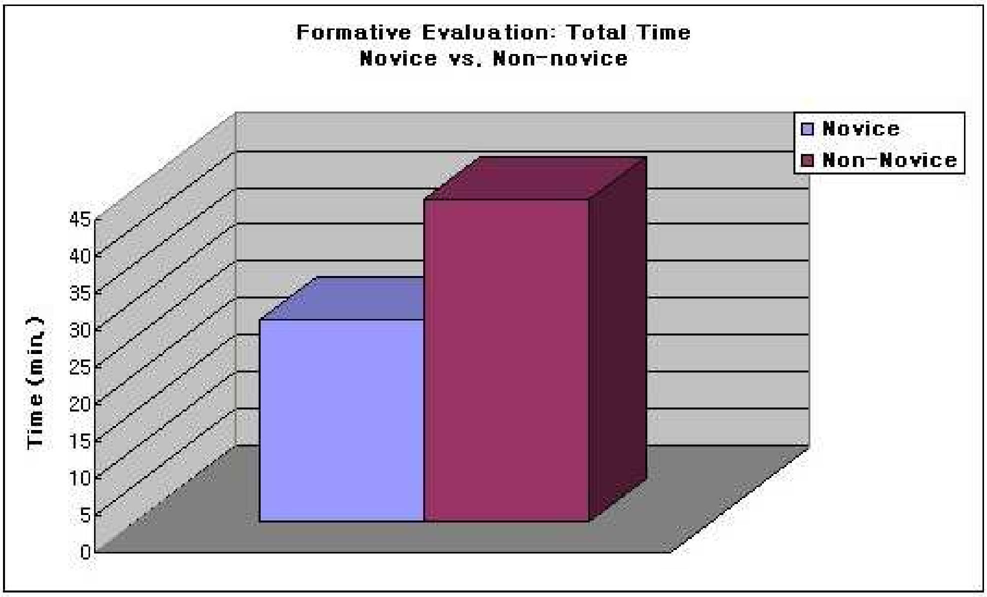


Figure 5.21: Formative User Evaluation: Experiment Time

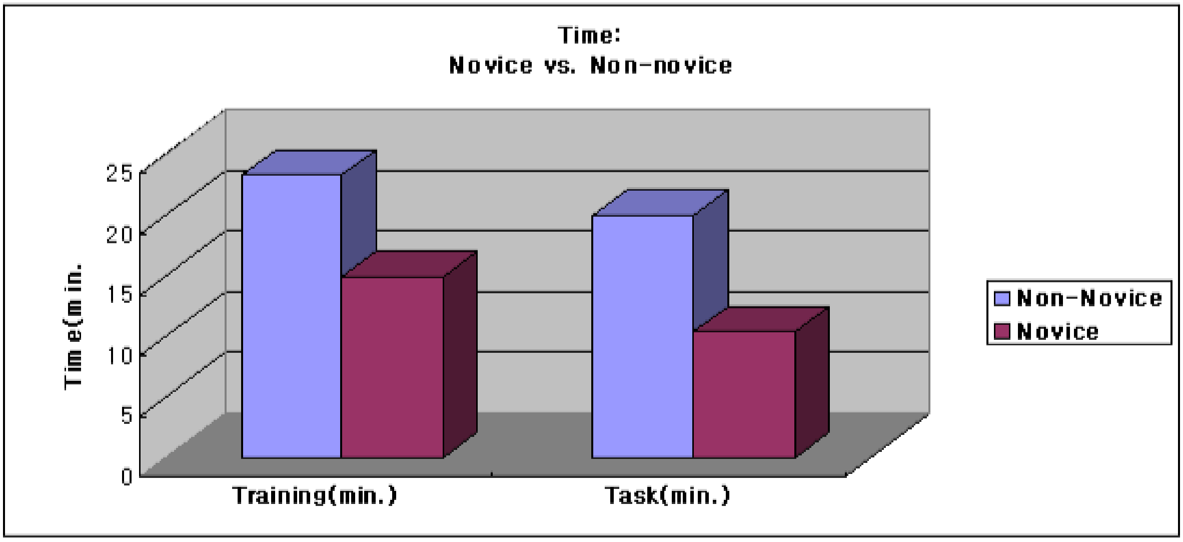


Figure 5.22: Formative User Evaluation: Training and Task Time

the user from context-switching between interaction space and display space, blocking her view of the inside box. However, because of the design of the input tools, users wanted to watch their handling of props, especially during the rotation and the scaling tasks. Because of this, we considered redesigning the current interface tool and/or the interaction space instead of the *ARBox*. Had we redesigned the current interface tools, we would have needed to generate new interaction techniques. Due to time constraints, we decided to make a new interaction space, *ARDesk*, where users can both watch their movement and observe the interaction outputs.

5.3 Summative Evaluation

As a result of our formative evaluation, we found that the design of interaction space, as well as the design of interface tools, has a large effect on task performance. Unlike *ARBox*, *ARDesk* may cause context-switching and interrupt the user's immersion in the CAVE^{TM} system. We anticipate that the new interaction space, *ARDesk*, will evince some of these same problems. The goal of our summative user evaluation was to observe the user's task performance in *ARDesk* compared to *ARBox*.

5.3.1 Participants

Looking at the results of our formative user evaluation, we can assume that our interface is more suitable for novices than for non-novices. We were supposed to recruit the same number of novices as non-novices, but due to time constraints, we recruited total 17 subjects, only five of whom were non-novices. They are all 18-50 years old.

5.3.2 Procedure

The procedure was identical to that of the formative user study because we had not redesigned the software test bed. Only the physical interaction environment was redesigned, nothing else being changed. Instead, we changed the post-questionnaire to get more specific answers from the subjects. All subjects underwent both the training and the task session. Figure 5.23, 5.24, 5.25, 5.26, and 5.27 show two sessions with *ARDesk*.

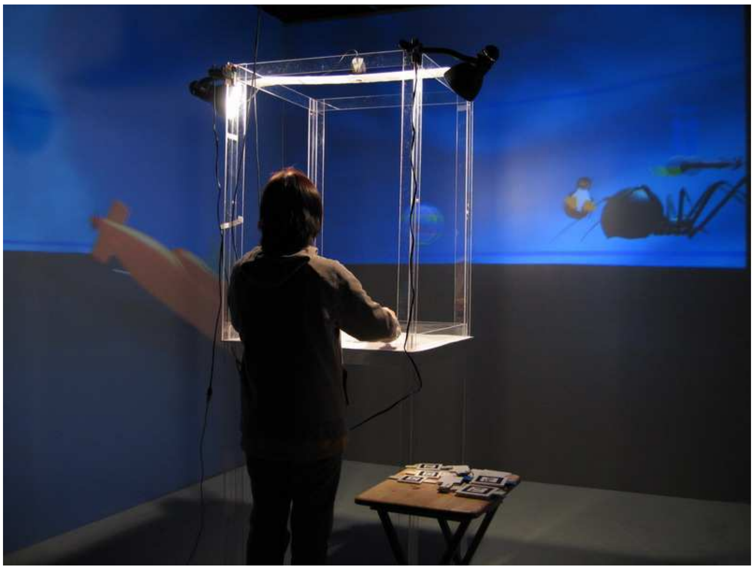


Figure 5.23: Training session on *ARDesk*

5.3.3 Results

In this study, we observed how users felt about our interface design, (i.e., prop design) and how much their task performances were affected by the new interaction space, *ARDesk*. For the former goal, we prepared more specific questions for the post-questionnaire after the experiment (see Appendix A). For the latter goal, we concentrated on time measurement, dividing subjects into novices and non-novices and examining the results. The results are as follows:

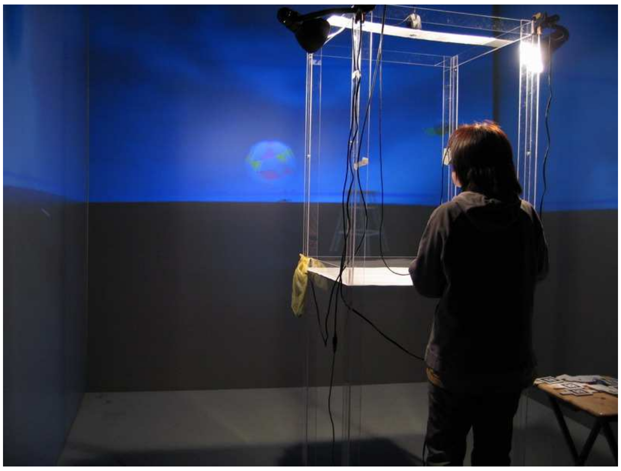


Figure 5.24: *Task 1*

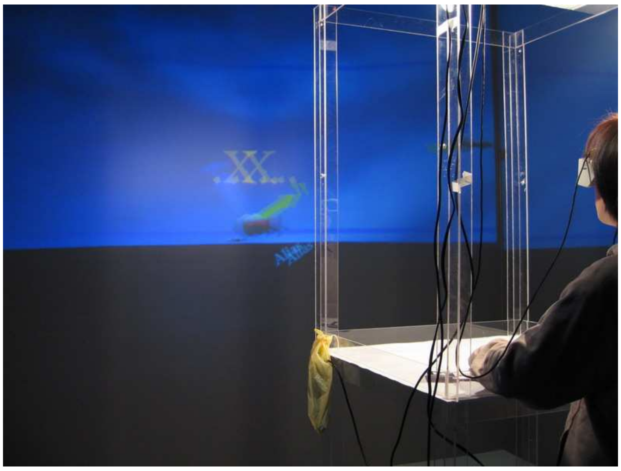


Figure 5.25: *Task 2*

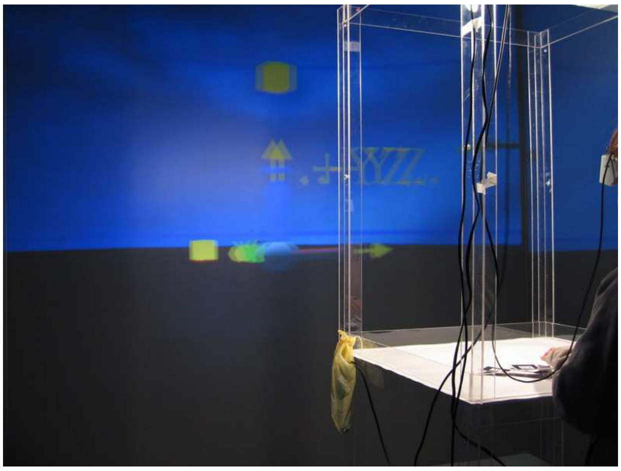


Figure 5.26: *Task 3*

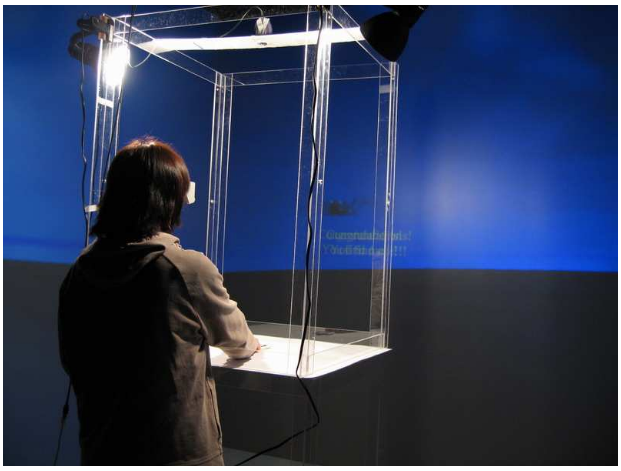


Figure 5.27: *Task 4*

- **Comfortableness**

  As Figure 5.28 shows, overall ratings of the translation task and the navigation task are higher than for the others. In the translation task, experienced users felt a bit more comfortable because they thought that the use of the prop was easier than the use of a wand would have been. Generally, users have to be specially trained to use selection and the translation techniques with 3D input devices. However, our object selection and translation techniques do not require any special training. On the contrary, in the navigation task, novices felt more comfortable because our navigation technique, designed in a different way from wand-based interfaces, is easier, although the difference confused experienced users. Experienced users also indicated that they could have gotten better haptic feedback from a wand-based navigation technique, pressing the wand-mounted joystick forward or backward .

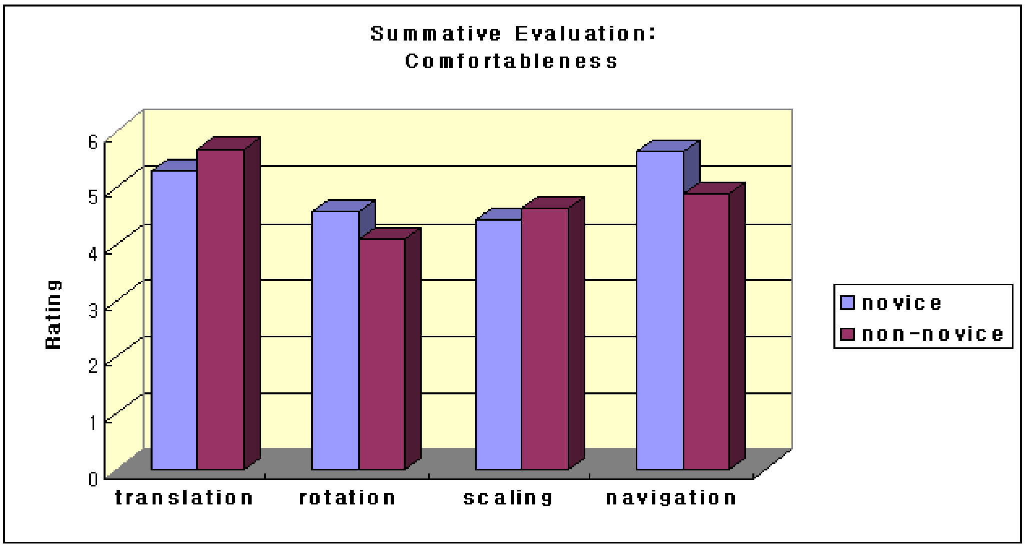


Figure 5.28: Summative User Evaluation: Comfortableness

- **Affordance**

  This subjective measurement is to assess the effect of our pattern design on a prop. In our interface system, no 3D image is superimposed on the prop. Users can see the actual pattern and map the pattern they see to the virtual object. For this reason, pattern design must also be considered with regards to affordance affecting the user's ability to map patterns properly.

As Figure 5.29 shows, all of tasks are highly rated by novice users, but not by non-novice users. Because the pattern size was limited, we designed the pattern with just two letters. Some users claimed a full word would have given better affordances. One subject suggested the use of a graphical pattern to symbolize the actual task.

The current patterns are intentionally very simply, and we did not expect that these patters would provide particularly good affordances; instead we wanted to examine the effect of the patterns designed. We found, however, that it could help users perform their task in the virtual world if the pattern were more relevant to the task, or at least to the virtual image.

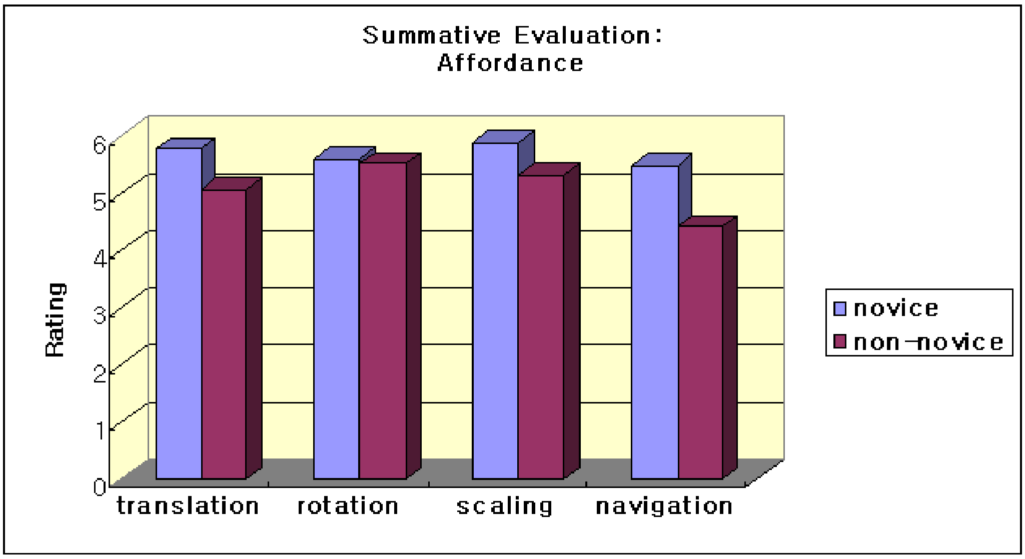


Figure 5.29: Summative User Evaluation: Affordance

- **Effectiveness**

We asked users if the prop and the test environment designs were appropriate and adequate to complete each task. They said that each task design itself was quite suitable, except for the navigation task: but they thought that the prop design for the rotation and the scaling tasks felt unnatural.

We assumed that the navigation task was very simply designed, and we did not put a lot of landmarks in the virtual world. However, some subjects failed to control the navigation speed, and, when they lost their direction, they could not get back to where they were before, and therefore not complete the mission. Thus, some of them indicated that the virtual space should be redesigned to include useful landmarks.

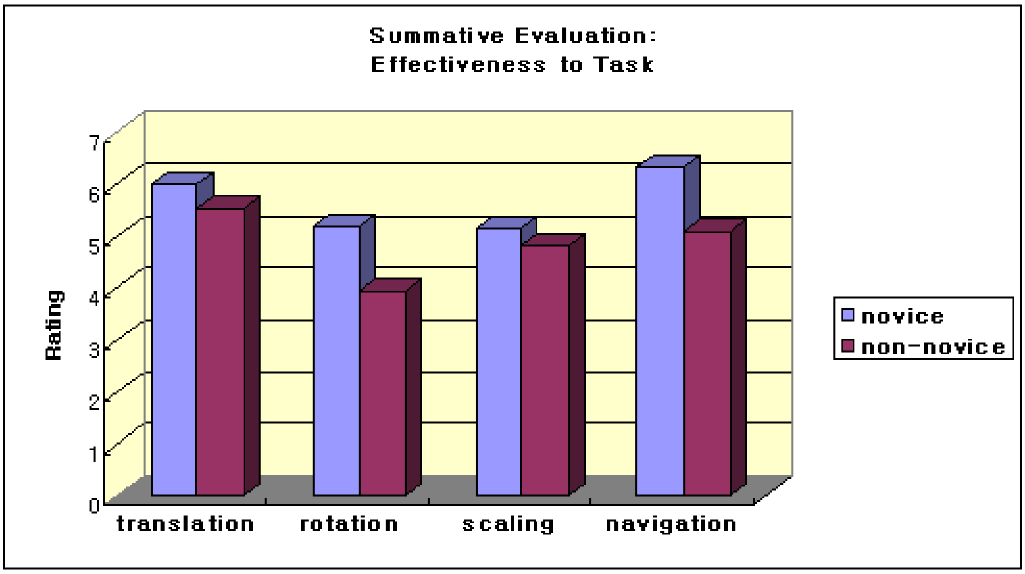


Figure 5.30: Summative User Evaluation: Effectiveness to Task

- **Intuitiveness**

Users indicated that our interface for the rotation and the scaling task was not sufficiently intuitive. In the rotation task, they would have preferred to rotate the object prop itself, without the use of any additional tool. In the scaling task, most of subjects had trouble controlling the scaling prop because of sensitivity required to control the distance between the object and the scaling prop. In addition, they would have preferred to rotate and scale the virtual objects more freely, without being limited to the three axes.

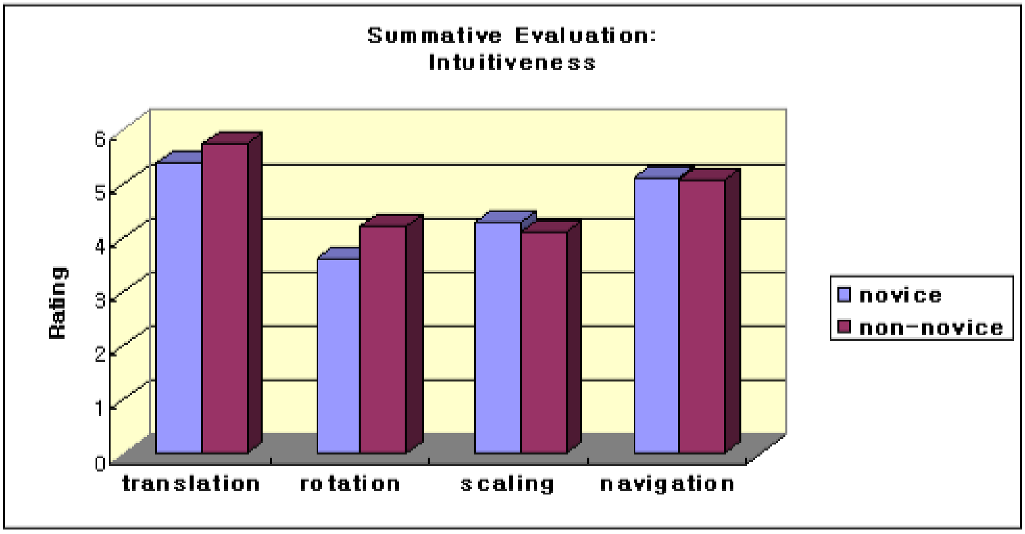


Figure 5.31: Summative User Evaluation: Intuitiveness

- **Immersion**

  We also asked about aspects of the physical environment that might have reduced the user's immersion in the virtual world. Five subjects indicated that the plastic glass frames occluded their visions. Four subjects replied that the two extra lamps were too bright, that the stereoscopic glasses were too heavy, and the cables tethered connected to the glasses interfered with their visions. Because of shadows, the AR system frequently produced detection errors, and some of our subjects were bothered by this. However, overall, the subjects said that physical factors did not significantly affect immersion.

- **Completeness of the task**

  Most of subjects said that they had no problem completing the translation and the navigation tasks. However, as aforementioned, they said that two techniques developed for the rotation and the scaling were quite difficult due to their prop designs. Nevertheless, they responded that each task itself was well designed for rapid completion.

We observed that most subjects finished their training session earlier with *ARDesk* than with *ARBox*. As Figure 5.32 shows, subjects spent a little more time in *ARDesk* than in *ARBox* for the training session. However, they could finish their tasks with *ARDesk* very quickly. The big difference between the *ARBox* and the *ARDesk* has to do with whether or not users can see their movements. When users can see their movements, their task performance is improved. Even though there is context-switching, and this may reduce the user's overall performance, this reduction is trivial.

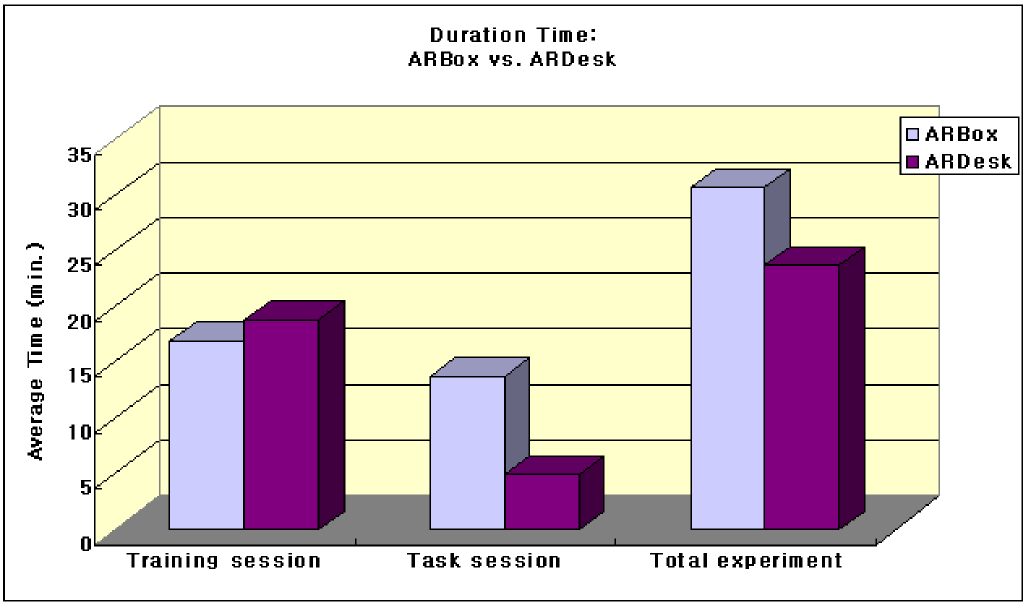


Figure 5.32: Duration Time: *ARBox* vs. *ARDesk*

5.4 Discussion

We designed the software test bed with the intention that participants would be excited in doing their tasks during the experiment. Since participants had to perform four different interaction techniques for different tasks, we needed to make each task as easy and simple as possible. However, the tasks had to be appropriate to the effective examination of our interface. Most of the participants performed well, and they were very excited in doing their tasks. Some subjects even shouted with satisfaction whenever they completed each task. We believe that our test design for each task experiment was well made.

Since *ARDesk* is unblocked from the outside world, unlike *ARBox*, it is easy inadvertently to create shadows of hands and props. These shadows caused detection errors, and this led frequently to the the evaluator's having to restart each task a bit more often than we would have liked. Nevertheless, users' task performance was much better in *ARDesk*. Thus, we assumed that task performance might have been improved had we measured usability differently.

We still have usability issues with *ARDesk*, such as the plastic frame occlusion and bright lamps, as well as with the prop design. Most of subjects had trouble controlling props with two hands in the rotation and the scaling tasks. They wanted to manipulate these control props with one hand. Our original idea was to design control props easily handled with one hand, and we thought that we had done so. However, most subjects had already started using two hands before the evaluator could suggest that they use only one.

In addition, our subjects had anticipated more intuitive ability to rotate or scale virtual objects. The current design allows for manipulation of virtual objects along only one axis, or along composite axes. Our subjects, however, wanted to rotate the virtual object more freely, as if it were a physical object, without any consideration of the rotation axis.

For future work, we first need to make the current system more reliable with regards to subject immersion. Immersion was extremely hindered by detection errors and the unforeseen, such as scenes coming up unexpectedly. Finally, we should significantly consider redesigning the current interface tools to improve the interaction techniques with them.

Chapter 6

Conclusion and Future Work

In this chapter, we summarize a new 3D user interface for an immersive virtual environment, which we proposed, address contributions, and then describe the direction for future work.

6.1 Summary

We presented a vision based TUI implemented using ARToolKit in the CAVE^{TM} system. This work consists of the interface system, which we designed, and the interaction techniques to prove that our approach is suitable for fundamental tasks in VEs.

Our interface system is comprised of three subsystems, as follows,

- **AR system**

  This subsystem is of the AR part in our interface system. The AR system includes a camera to track everything that happens in the real space, i.e., in the work area, the network communication system, and the ARToolKit. On this subsystem, we developed our interface application to get tracking information and deliver this information to the VR part, the user's interaction information as well as the tracking information.

Our interface tools serve this subsystem as input devices, such as the wand device.

- **Graphic system**

  As part of the VR system, the Graphic system is responsible for generating the virtual world with which the user interacts. We developed our graphic application to create the virtual scene with virtual objects manipulated by users. This application required the conversion of the difference coordinate systems and the different scaling units to generate the virtual scene correctly with the tracking information sent from the AR system. Socket communication and the shared memory functionality enabled us to transfer tracking information over the network link.

- **CAVE system**

  Our interaction space is placed in the CAVE^{TM} room, which is equipped with three 10 x 10 foot projection screens and a floor, stereoscopic glasses, and a head tracker. Users can immerse themselves in the virtual world with this system environment, doing their virtual tasks while watching the 3D virtual scene.

Our initial interface system used toy blocks as input tools (i.e., props). A user stands on the floor at the worktable. Props are put on the table, and the camera is placed over the worktable. When we turned off the ambient light in the lab, and only the projectors were turned on, the camera could not detect any of the marker on the props. In our next attempt, we applied the *White Rectangle*, a white rectangular image projected on the floor. It was bright enough for the camera to detect the markers, but its light was too shiny and reflective for the camera to recognize them. Furthermore, we encountered another problem using the *White Rectangle* because the top projector of VT-CAVE^{TM} was broken. Our alternative solution was to add an extra light bulb and put the camera directly over the worktable. This bulb gave enough brightness for the camera to recognize patterns. We performed the pilot study with this system design and got good feedback from the subjects (see Chapter 5).

Since it was one of our primary concerns that users be able to manipulate props comfortably in our interface system, the experiment designed to test our interface had to be carefully considered also. It had to be done simply and as quick as possible because of time-constraints. We went rapidly through several trials to devise a better low-cost design. Thus, our *ARBox* was born, and it was used for formative user study.

ARBox, made out of a big packing box, is quite large to prevent users from both moving out of the camera frame and making shadows with physical objects including their hands. We can also make sure that the light strength is consistent.

We redesigned our props in keeping with both the camera's frame size and the *ARBox*'s size. To enlarge the interaction space within the camera's frame, we discovered that props had to have the following properties:

- They had to be designed so as to maintain the greatest possible distance between the camera and the markers on the props.
- They had to be designed with as small as possible: for example, a 6x6 (cm) marker is better than an 8x8 marker, although the detection performance is much higher in 8x8 markers.
- They had to be easily grabbed and manipulated by the user.

Observations made during the pilot study and the formative user evaluation indicated that prop-based TUI appealed subjects because of its ease of use, intuitiveness, and comfortableness. After we conducted the formative user evaluation, we redesigned our interaction space and made the *ARDesk*. We tried to reduce occlusion of the user's view by using transparent plastic glasses in the *ARDesk*. Although the *ARDesk*'s columns somewhat undermined the user's immersion in the VE, task performance was better in this open interaction space than in the closed interaction space of the *ARBox*.

We assumed that our interface design for the rotation and scaling tasks might be hard to control precisely. Therefore, we designed four experiment tasks which were effective but simple enough for the user to complete with the current interface. However, during the training session, we observed that the some of subjects wanted to be able to manipulate the rotation and scaling tasks more precisely. They were not satisfied with the interaction techniques for these two tasks because they could manipulate only along the axis. They wanted to manipulate virtual objects without having to worry about the axis.

For the training session, our software test bed provided four virtual objects, and the users could freely manipulate them for four tasks. For the task session, we designed four experiment tasks that the participants had to complete. We observed that our software test bed for the navigation task lacked landmarks to indicate to the user how to find the right direction and how to return to the point of origin when they became lost. Since we had simply borrowed the navigation technique from a general joystick-based interface, we were not surprised when it turned out to be not entirely suitable for the 3D interaction technique. Nevertheless, most subjects were enthusiastic about the navigation task as a result of our having designed the task to be entertaining and to give the users some sense of accomplishment.

6.2 Contribution

The following is a list our contributions to the 3D user interface for semi-immersive virtual environments, especially CAVE-like projection-based large display environments.

- We have proposed for the first time a Tangible and Video-based 3D interface system for the CAVE^{TM} immersive virtual environment.
- We developed our own props and interaction space for evaluating our approach.
- We presented novel interaction techniques for fundamental 3D interaction tasks (i.e., object selection, translation, rotation, and scaling) and for the system control task.
- We demonstrated that it is possible for users to interact with the virtual world using tangible props without any extra electronic or electromagnetic devices, the vision

technique being used only for tracking.

- We provided a list of possible applications and interaction techniques using our interface that can be realized with props and interaction space more advanced from the current design.

We believe that our interface system is a very interesting and challengeable approach for both Virtual Reality and Augmented Reality. Little work has been done using video-based 3D interface in CAVE^{TM} systems. To the best of our knowledge, ours is the first study of tangible user interface for CAVE^{TM} applications utilizing video-based AR techniques.

This interface system is an attempt to develop a practical video-based perceptual user interface. Although the video-tracking implementation leaves much room for further improvement, the work that we have done is exciting with regards to the prospect for future development of natural user interfaces in VEs. Finally, the current interaction space can be reused as is in future studies of interaction techniques using video-based TUIs.

Despite some usability issues, such as limitation of the work space or context-switching between interaction and display space, as well as unreliable system issues, we feel that our work still casts significant light on the future design of research in TUI-based interaction techniques.

We hope that our work will inspire more research rigorously undertaken in the hopes of advancing tangibly the use of vision-based 3D interfaces in semi-immersive virtual environments.

6.3 Future Direction

Since our approach has just started, there remain many open issues, as well as challenges. This thesis finishes with a look at possible future directions our research might take by listing current issues deserving future attention.

1. **Redesign of props and interaction space.**

   From the results of our user studies, we feel that our design can be improved to enhance both the task and system performance. For example, the props can be more delicately designed and the test bed more robust.

   - **Props**

     In the current study, most users had to use two hands to indicate the direction for rotating or scaling objects. If the paddle prop were designed, as in Figure 6.1, users could more easily control which axis marker is hidden or shown. In near future work, we will redesign the paddle prop and perform a new user evaluation. We also need to devise a new prop design to give users more freedom of direction in the rotation and scaling tasks.

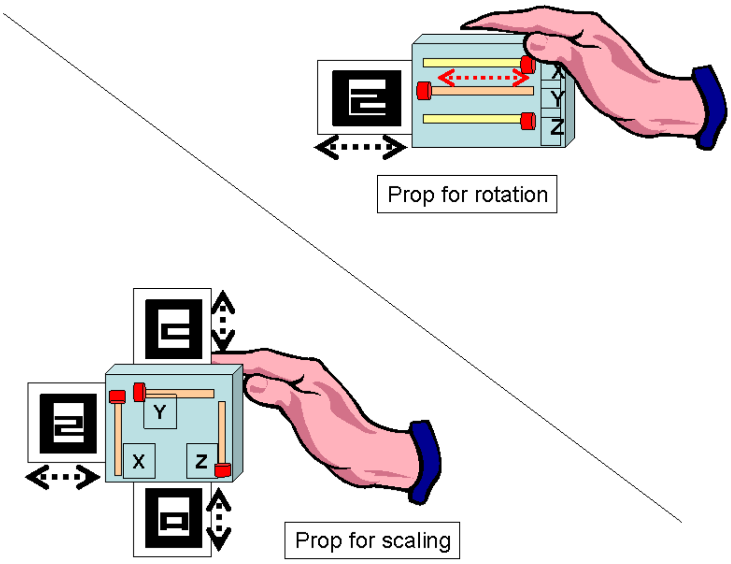


Figure 6.1: Props design for future work

   - **Interaction Space**

     As we mentioned earlier, we need to consider reducing the gap between interaction space and display space. Our first solution was to use the big box, in which users cannot see their interactions, blocking unnecessary access from the outside space. The problem with this box, however, was that it undermined the user's view of the front screen wall, as well as preventing the user from seeing her own physical interactions. Another approach was to use transparent plastic glass to enable the user to see any view on the screens, even through interaction space. We might also add flexibility by adjusting the camera's position when using different types of camera. Although we got the quite good task performance with the *ARDesk* made of transparent plastic glass in the summative user evaluation, we found that we needed to remove two extra lamps from the *ARDesk*. If we can design a new prop containing self-light or that is wrapped with neon paper sufficient to be detected by a camera, we will be able to improve the current interaction space by enhancing user immersion.

- **Use of Personal Digital Assistant (PDA) as a marker**

  If we do not need to manipulate multiple objects at once, we can use PDA instead of the paper-based props. An experiment [24] performed to examine the effect of different marker materials used a transflective LCD of PDA instead of paper-based markers to achieve dynamic changes in the markers. If our current control markers were replaced with PDA, we too might be able to enhance performance, since PDA provides for a lot of interesting functionalities, such as sound effect, button manipulation, and touch screen.

Observations made during our user studies indicated that our prop-based interface is easily understood and used by participants in the project, particularly when they are novices unfamiliar with 3D interfaces in immersive VEs. As regards tangible input devices, it is significant that we consider how we might design props to provide users with good affordances. Were our interaction space more carefully designed, it might provide a more facile interface in which to perform fundamental tasks in a more seamless way, even in the case of separation of interaction space and display space.

2. **Development of various interaction techniques**

   We developed interaction techniques to investigate how efficient our new approach was when applied to a semi-immersive virtual environment. Given the rich interaction techniques based on this interface, it would be still more interesting to explore other relevant applications that take advantage of the vision-based tangible objects' affordances. As we have shown in this thesis, a straightforward application is one that directly uses the prop's 3D information to manipulation the 3D virtual object and navigate in the 3D virtual world. The interaction techniques presented in this research may not allow the user to manipulate virtual objects in far space, i.e., out of arm's reach. In our work in the near future, we will improve the current interaction techniques to enable the user to grab and release virtual objects in far space just as the user now grabs and releases virtual objects in near space. This is possible without the use of any enhanced interaction technique, such as Go-Go [37] or HOMER [7], because all 3D tracking information exists only within the camera frame. When the camera loses the prop-based 3D information, the virtual object ceases to move. Then, when the prop enters the frame again, the virtual object starts moving from the place where it was along the prop's movement. This technique is similar to the mouse-based interaction technique on desktop computer. While the current interaction technique directly maps the prop's position data to the virtual object, future interaction technique will map the prop's relative position.

3. **Applying for collaboration**

   A possible application can be implemented to support collaborative work. As we have shown, our interface has two separated spaces for interaction and display. This can be especially beneficial in some applications.

   For example, imagine that someone needs to be trained for an assembly task. She seeks the advice of an assembly expert. She stands on the floor of the CAVE^{TM} room, where she is trained in the task. She uses some common input device, depending to the task.

> Two experts sit outside the CAVE^{TM} room. They guide the trainee, talking to her or manipulating virtual objects using props on the *ARDesk*. In this case, display space would be for the trainee and interaction space for two experts. They watch what the trainee is doing, give advice directly, and show her what they want her to do by the direct manipulation of virtual objects.

In future work, it will be of value to conduct more extensive user studies with the most advanced system designs, including more rigorous measurements, both objective and subjective. It is thought-provoking to compare our interface based interaction techniques with other interface based interaction techniques for the CAVE^{TM} applications, to find out the advantages and disadvantages of our interface system in different applications. With regards to usability issues, it is desirable that we go back and test our interface system iteratively, and then improve it. We hope by these means to improve performance in really practical situations by collecting more and more feedback from participants in future studies.

Bibliography

[1] D. Anderson, J. Frankel, J. Marks, A. Agarwala, P. Beardsley, J. Hodgins, D. Leigh, K. Ryall, E. Sullivan, and J. Yedidia. Tangible interaction + graphical interpretation: A new approach to 3d modeling. In *In Proceedings of ACM SIGGRAPH 2000*, pages 393–402, New York, 2000. ACM Press // ACM SIGGRAPH.

[2] J. Anstey, H. Hörtner, D. Cox, D. Sandin, P. Sermon, and J. Shaw. The cave and beyond: Vr art in museums and galleries. SIGGRAPH 2001 online Panels.

[3] R. Azuma. Making direct manipulation work in virtual reality. *Course notes on "Registration" and "Correcting for Dynamic Error" from Course Notes 30, ACM SIGGRAPH '97*, August 1997.

[4] R. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[5] S. Baldis. Input device for 3d interactions. Empirical studies of the Effectiveness of Input Devices in 3D Spaces.

[6] M. Billinghurst, H. Kato, and I. Poupyrev. Collaboration with tangible augmented reality interfaces. In *HCI International 2001*, New Orleans, LA, USA, August 2001.

[7] D. Bowman and L. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *In Proceedings of the Symposium on Interactive 3D Graphics*, pages 35–38, 1997.

[8] D. Bowman, E. Kruijff, J. LaBiola, and I. Pourpyrev. *3D User Interfaces:Theory and Practice*. Addison-Wesley, 2004.

[9] D.A. Bowman, C.A. Wingrave J. Chen, J. Lucas, A. Ray, N.F. Polys, Q. Li, Y. Haciahmetoglu, J. Kim, S. Kim, R. Boehringer, and T. Ni. New directions in 3d user interfaces. Under submission to ACM Transactions on Computer Human Interaction (05TOCHI), 2005.

[10] W. Broll, E. Meier, and T. Schardt. The virtual round table - a collaborative augmented multi-user environment. In *Proceedings of the ACM CVE 2000*, pages 39–46, New York, 2000. ACM.

[11] K. Camerata, E.Y. Do, M.D. Gross, and B.R. Johnson. Navigational blocks: Tangible navigation of digital information. In *ACM Conference on Human Factors in Computing Systems*, pages 751–752, Minneapolis , Minnesota, USA, April 2002.

[12] J.E. Carvalho, J. Luzio, P. Santos, M. Dias, and J. Jorge. Usability evaluation of tangible user interfaces for augmented reality. In *IEEE International Augmented Reality Toolkit Workshop*, Tokyo, Japan, October 2003.

[13] InterSense Company. http://www.isense.com/products/prec/is900/wireless.htm, Web Page.

[14] P.J. Costello. Health and safety issues associated with virtual reality. Technical Report 37, a review of current literature, 1997.

[15] C. Cruz-Neira, J. Leigh, M. Papka, C. Barnes, S. Das S. Cohen, T. Roy L. Siegal C. Vasilakis T. DeFanti R. Engelmann, R. Hudson, and D. Sandin. A report on visualization applications in the cave virtual reality environment. Technical report, IEEE Computer Graphics, 1993.

[16] C. Cruz-Neira, D. Sandin, and T. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *In SIGGRAPH 93 Conference Proceedings*, volume 27, pages 135–142, 1993.

[17] A. Van Dam. Post-wimp user interfaces: The human connection. In *Communication of the ACM*, volume 40, pages 63–67, 1997.

[18] M. Fjeld, N. Ironmonger, S. Guttormsen Schar, and H. Krueger. Design and evaluation of four ar navigation tools using scene and viewpoint handling. In *In Proceedings of INTERACT 2001*, pages 214–223, 2001.

[19] E.D. Guzman, F.W. Ho-Ching, T. Matthews, T. Rattenbury, M. Back, and S. Harrison. Eewww!!: tangible interfaces for navigating into the human body. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 806–807, New York, NY, USA, 2003. ACM Press.

[20] K. Hinckley, R. Pausch, D. Proffitt, and N. F. Kassell. Two-handed virtual manipulation. *ACM Trans. Comput.-Hum. Interact.*, 5(3):260–302, 1998.

[21] R. Holloway and A. Lastra. Virtual environments: A survey of the technology, 1993. Technical Report, University of North Carolina, Chapel Hill, NC.

[22] H. Ichida, Y. Itoh, Y. Kitamura, and F. Kishino. Interactive retrieval of 3d shape models using physical objects. In *Proceeding of the ACM Multimedia*, pages 692–699, NY, USA, Oct 2004.

[23] S.K. Isabelle, R.H. Gilkey, R.V. Kenyon, G. Valentino, J. Flach J, and T.R. Anderson C. Spenny. Defense applications of the cave (cave automatic virtual environment). In *Proceedings of SPIE: 11th Annual Conference on Aerospace/sensing simulation and control*, Orlando, Fl, April 1997.

[24] S. Jang, S. Lee, and W. Woo. cpost-it: Context-based information sharing system, 2003. Lecture Note Computer Science.

[25] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *In Proceedings of Int. Symp. on Augmented Reality 2000*, 2000.

[26] H. Kato, K. Tachibana, M. Tanabe, T. Nakajima, and Y. Fukuda. A city-planning system based on augmented reality with a tangible interface. In *ISMAR 2003*, 2003.

[27] D.F. Keefe, D.A. Feliz, T. Moscovich, D.H. Laidlaw, and J.J. LaViola. Cavepainting: A fully immersive 3d artistic medium and interactive experience. In *in Proceedings of 2001 Symposium on Interactive 3D Graphics, ACM SIGGRAPH*, 2001.

[28] G.A. Lee, M. Billinghurst, and G.J. Kim. Occlusion based interaction methods for tangible augmented reality environments. In *In proceedings of the ACM Siggraph International Conference on Virtual Reality Continuum and its Applications in Industry (VRCAI 2004)*, pages 419–426, June 2004.

[29] J. Leigh, A. Johnson, T. DeFanti, and M. Brown. A review of tele-immersive applications in the cave research network. In *Proc. IEEE Virtual Reality*, 1999.

[30] B. Lok and L. Hodges. *Human-Computer Interaction in Virtual Reality.* Berkshire Encyclopedia of Human-Computer Interaction, 2004.

[31] L. Kok-Lim Low, I. Adrian, W. Greg, and L. Anselmo. Combining head-mounted and projector-based displays for surgical training. In *IEEE Virtual Reality Conference 2003 (VR'03)*, 2003.

[32] J. F. Lucas, J. Kim, and D. A. Bowman. Resizing beyond widgets: object resizing techniques for immersive virtual environments. In *CHI Extended Abstracts 2005*, pages 1601–1604, 2005.

[33] S. Malik. Robust registration of virtual objects for real-time augmented reality. Master's thesis, Carleton University, Ottawa, Ontario, Canada, May 2002.

[34] T. Mazuryk and M. Gervautz. Virtual reality history, applications, technology and future, 1996. Technical Reports at the Institute of Computer Graphics and Algorithms, Visualization and Animation Group, Vienna University of Technology.

[35] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. *Proc. Telemanipulator and Telepresence Technologies of SPIE*, December 1995.

[36] W. Piekarski. *Interactive 3d modelling in outdoor augmented reality worlds.* PhD thesis, University of South Australia, Division of Information Technology, Engineering, and the Environment, 2004.

[37] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. Go-go interaction technique: Non - linear mapping for direct manipulation in vr. In *ACM UIST'96*, pages 79–80, 1996.

[38] R. Raskar, G. Welch, and W-C. Chen. Table-top spatially-augmented realty: Bringing physical models to life with projected imagery. In *Proc. 2nd International Workshop Augmented Reality (IWAR 99)*, pages 64–71, Los Alamitos, California, 1999. IEEE CS Press.

[39] M. Rauterberg, M. Fjeld, H. Krueger, M. Bichsel, U. Leonhardt, and M. Meier. Build-it: a planning tool for construction and design. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*, pages 177–178, New York, NY, USA, 1998. ACM Press.

[40] A.F. Seay, D. Krum, L. Hodges, and W. Ribarsky. Direct manipulation on the virtual workbench: Two hands aren't always better than one. Technical report, Graphics, Visualization, and Usability Center - Georgia Institute of Technology, 2000.

[41] G. Smith and W. Stuerzlinger. On the utility of semantic constraints. In *Eurographics workshop on Virtual Environments 2001*, 2001.

[42] I. Sutherland. The ultimate display. In *Proceedings of IFIP*, volume Congress(2), pages 506–509, 1965.

[43] B.H. Thomas, H. Slay, and R. Vernik. Tangible user interaction using augmented reality. In *Third Australasian User Interface Conference (AUIC2002)*, 2002.

[44] C. Ulbricht and D. Schmalstieg. Tangible augmented reality for computer games. Technical report, The 3rd IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2003), September 2003.

[45] J. Vallino. *Interactive Augmented Reality.* PhD thesis, University of Rochester, Rochester, NY, 1998. PhD Thesis.

Appendix A

Questionnaires

A.1 Pilot Study

Tangible User Interface for CAVE based on Augmented Reality Technique

Subject# ____

Pilot Study – Post Experiment Questionnaire

After Experiment

Question P4-6: Please rate the techniques according to your personal preference. Please Circle your answer.

P4 Selection Did not like 1-----2-----3-----4-----5-----6-----7 Liked A lot

P5 Manipulation Did not like 1-----2-----3-----4-----5-----6-----7 Liked A lot

P6 Navigation Did not like 1-----2-----3-----4-----5-----6-----7 Liked A lot

Question P7-9: Please rate the techniques according to how well you could control the virtual object. Please Circle your answer.

P7 Selection Hard to control 1-----2-----3-----4-----5-----6-----7 Easy to control

P8 Manipulation Hard to control 1-----2-----3-----4-----5-----6-----7 Easy to control

P9 Navigation Hard to control 1-----2-----3-----4-----5-----6-----7 Easy to control

Question P10-12: Please rate the techniques according to how fast you were able to complete the tasks. Please Circle your answer.

P10 Selection Very slow 1-----2-----3-----4-----5-----6-----7 Very Fast

P11 Manipulation Very slow 1-----2-----3-----4-----5-----6-----7 Very Fast

P12 Navigation Very slow 1-----2-----3-----4-----5-----6-----7 Very Fast

Question of Tiredness: P13-15 Please rate the techniques according to how tiring the techniques were. Please Circle your answer.

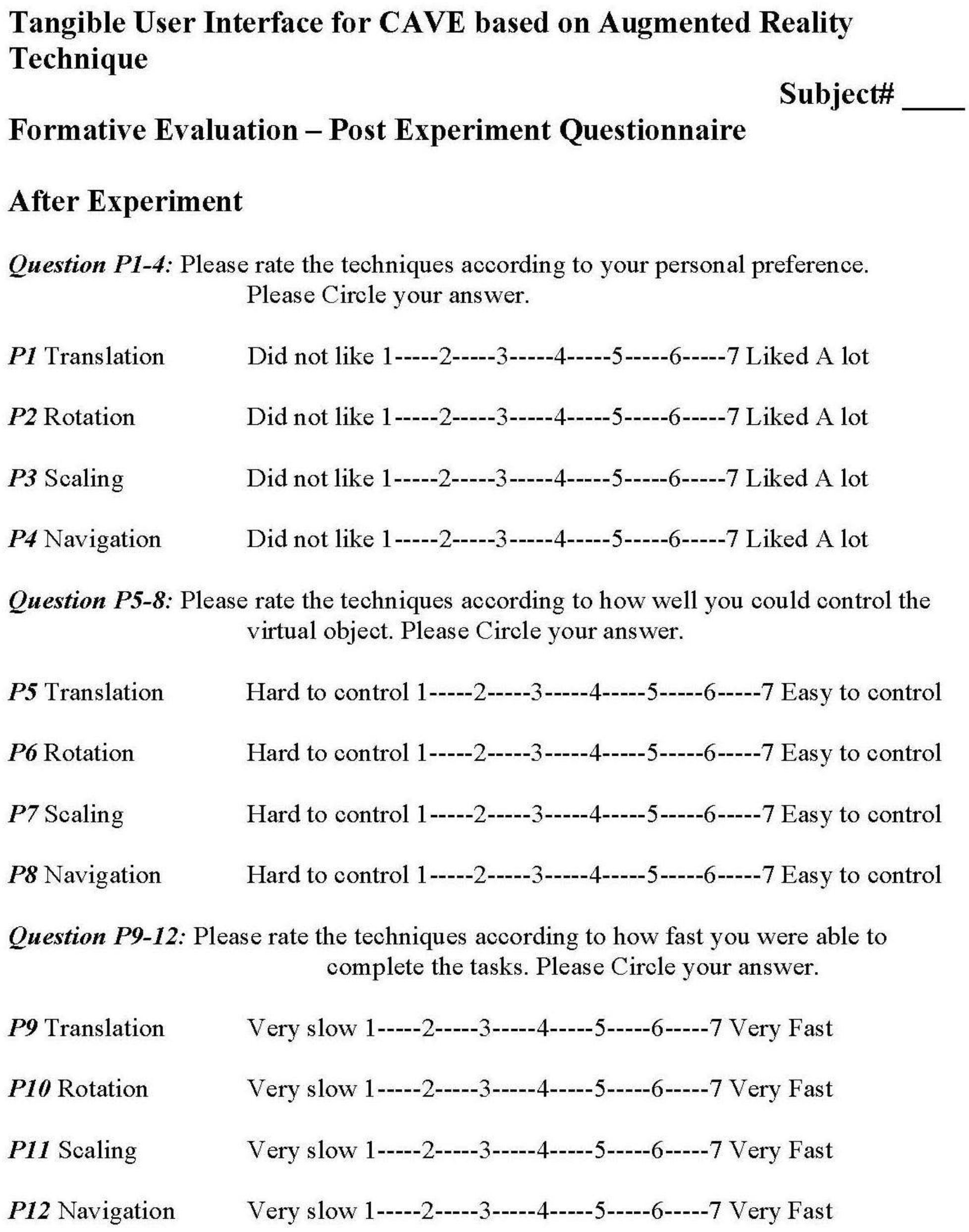
P13 Selection Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P14 Manipulation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P15 Navigation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

Thank You!

A.2 Formative User Evaluation



Tangible User Interface for CAVE based on Augmented Reality Technique

Subject# ____

Formative Evaluation – Post Experiment Questionnaire

After Experiment

Question P1-4: Please rate the techniques according to your personal preference. Please Circle your answer.

P1 Translation Did not like 1-----2-----3-----4-----5-----6-----7 Liked A lot

P2 Rotation Did not like 1-----2-----3-----4-----5-----6-----7 Liked A lot

P3 Scaling Did not like 1-----2-----3-----4-----5-----6-----7 Liked A lot

P4 Navigation Did not like 1-----2-----3-----4-----5-----6-----7 Liked A lot

Question P5-8: Please rate the techniques according to how well you could control the virtual object. Please Circle your answer.

P5 Translation Hard to control 1-----2-----3-----4-----5-----6-----7 Easy to control

P6 Rotation Hard to control 1-----2-----3-----4-----5-----6-----7 Easy to control

P7 Scaling Hard to control 1-----2-----3-----4-----5-----6-----7 Easy to control

P8 Navigation Hard to control 1-----2-----3-----4-----5-----6-----7 Easy to control

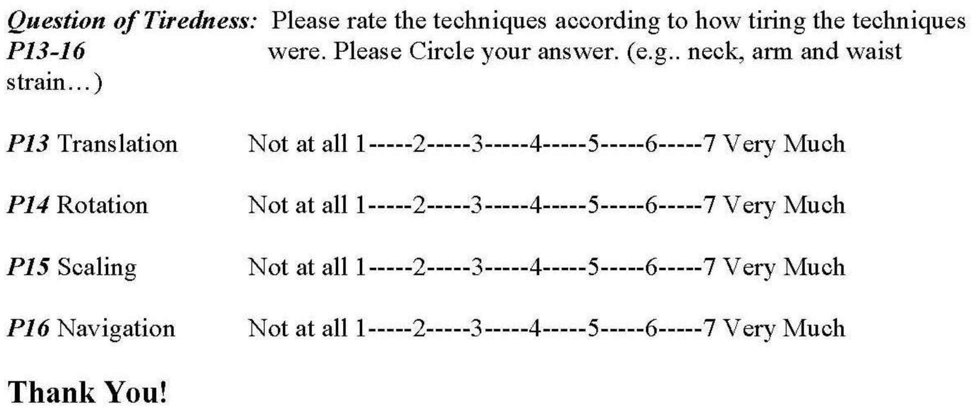
Question P9-12: Please rate the techniques according to how fast you were able to complete the tasks. Please Circle your answer.

P9 Translation Very slow 1-----2-----3-----4-----5-----6-----7 Very Fast

P10 Rotation Very slow 1-----2-----3-----4-----5-----6-----7 Very Fast

P11 Scaling Very slow 1-----2-----3-----4-----5-----6-----7 Very Fast

P12 Navigation Very slow 1-----2-----3-----4-----5-----6-----7 Very Fast



Question of Tiredness: Please rate the techniques according to how tiring the techniques
P13-16 were. Please Circle your answer. (e.g.. neck, arm and waist
strain…)

P13 Translation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P14 Rotation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P15 Scaling Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P16 Navigation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

Thank You!

A.3 Summative User Evaluation

Tangible User Interface for CAVE based on Augmented Reality Technique

Subject# ____

Summative Evaluation – Post Experiment Questionnaire

After Experiment

Question P1 : Please overall rate the experiment environment (i.e. work desk) according to how comfortable you feel during the experiment.

Did not like 1-----2-----3-----4-----5-----6-----7 Liked A lot

Additional Comment

Question P2-5: Please rate the techniques according to how comfortable you feel about handling props. Please Circle your answer.

P2 Translation Not at all 1-----2-----3-----4-----5-----6-----7 Very comfortable
(e.g. "O1", "O2", "O3" or "O4")

P3 Rotation Not at all 1-----2-----3-----4-----5-----6-----7 Very comfortable
(e.g. rotation prop(RX, RY, RZ))

P4 Scaling Not at all 1-----2-----3-----4-----5-----6-----7 Very comfortable
(e.g. scaling prop(SX, SY, SZ))

P5 Navigation Not at all 1-----2-----3-----4-----5-----6-----7 Very comfortable
(e.g. "Na")

Additional Comment

Question P6-9: Do you think patterns on the prop (e.g. *"O1", "O2", "O3" , "O4", Na, RX, RY, RZ, SX, SY and SZ*), give any meaning to you?
For example, even though the instructor does not ask to grab one or two of "O1", "O2", "O3" or "O4" props, can you assume that those patterns would be for virtual objects?

P6 Translation Not at all 1-----2-----3-----4-----5-----6-----7 Yes, absolutely.
(In case of "O1", "O2", "O3" or "O4"patterns, can you assume that those patterns would be for virtual objects?)

P7 Rotation Not at all 1-----2-----3-----4-----5-----6-----7 Yes, absolutely
*(In case of "**RX, RY, RZ**" patterns, can you assume that those patterns would be for rotating objects?)*

P8 Scaling Not at all 1-----2-----3-----4-----5-----6-----7 Yes, absolutely
*(In case of "**SX, SY, SZ**" patterns, can you assume that those patterns would be for scaling objects?)*

P9 Navigation Not at all 1-----2-----3-----4-----5-----6-----7 Yes, absolutely
*(In case of "**Na**" pattern, can you assume that those patterns would be for navigation?)*

Additional Comment

Question P10-13: Do you think those props and the test environment are suitable for each task? Please Circle your answer.

P10 Translation Not at all 1-----2-----3-----4-----5-----6-----7 Yes, quite well.

P11 Rotation Not at all 1-----2-----3-----4-----5-----6-----7 Yes, quite well

P12 Scaling Not at all 1-----2-----3-----4-----5-----6-----7 Yes, quite well

P13 Navigation Not at all 1-----2-----3-----4-----5-----6-----7 Yes, quite well

Additional Comment

Question P14-17: Please rate the techniques according to how intuitive this interface is for each task.

P14 Translation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P15 Rotation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P16 Scaling Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P17 Navigation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

Additional Comment

Question of Tiredness: Please rate the techniques according to how tiring the techniques
P18-21 were. Please Circle your answer. (e.g. neck, arm and waist strain…)

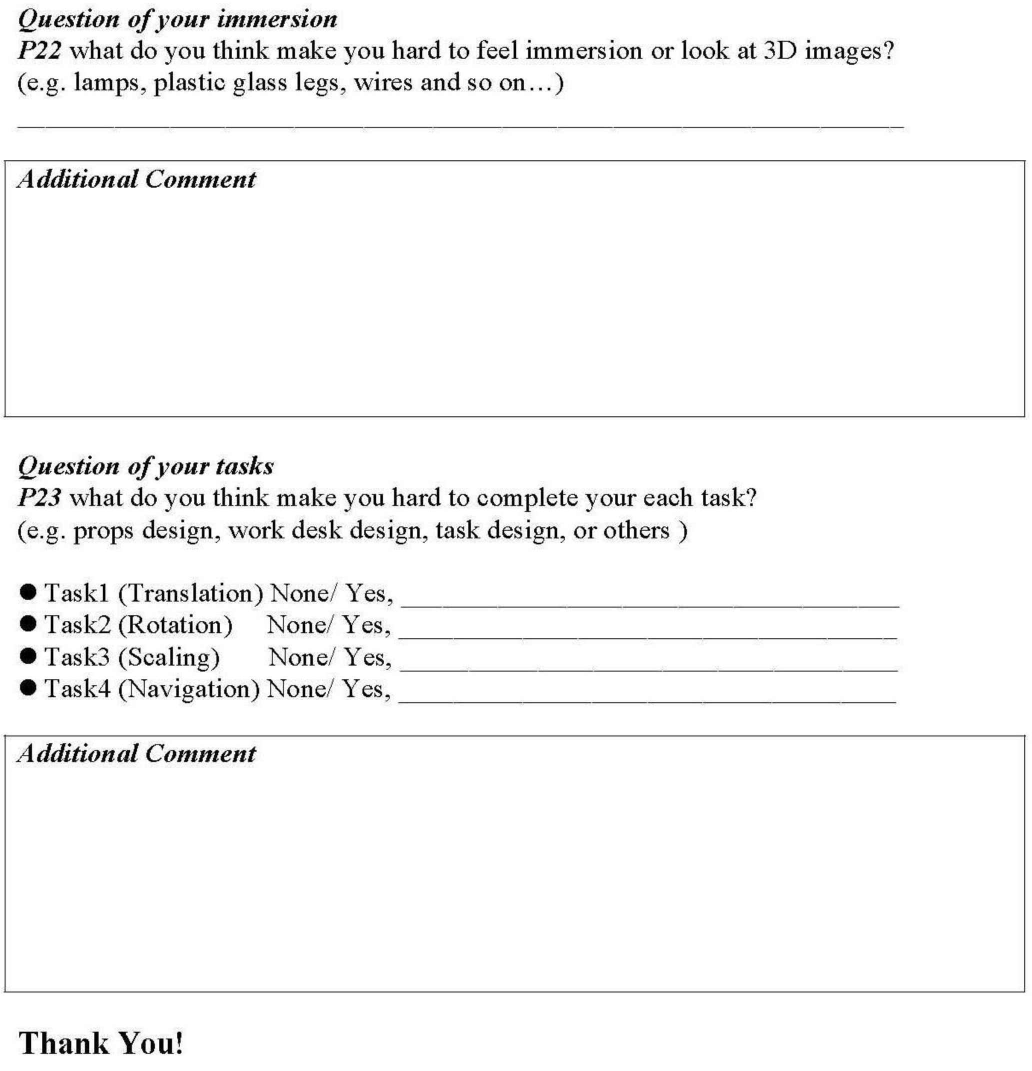
P18 Translation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P19 Rotation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P20 Scaling Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

P21 Navigation Not at all 1-----2-----3-----4-----5-----6-----7 Very Much

Additional Comment



Question of your immersion
P22 what do you think make you hard to feel immersion or look at 3D images?
(e.g. lamps, plastic glass legs, wires and so on…)
____

Additional Comment

Question of your tasks
P23 what do you think make you hard to complete your each task?
(e.g. props design, work desk design, task design, or others)

- ● Task1 (Translation) None/ Yes, ________________________________
- ● Task2 (Rotation) None/ Yes, ________________________________
- ● Task3 (Scaling) None/ Yes, ________________________________
- ● Task4 (Navigation) None/ Yes, ________________________________

Additional Comment

Thank You!

Vita

Ji-Sun Kim was raised in Seoul of South Korea where she spent six years to work in industry after graduating Sookmyung Women's University with a B.S. degree in 1996. She worked as a system programmer and developer for Samsung Electronics, Seoul Communication and ActozSoft companies and pursued here various skills and experiences. Following the admission from Virginia Tech for the 2003 fall semester, she moved to Virginia in the United States and currently is studying in Computer Science department at Virginia Tech. She is currently a candidate for a Master of Science degree in Computer Science.