Usability of 2D palmtop interaction device in Immersive Virtual Environments

by

Lewis Charles Hill II

A thesis submitted to the graduate faculty in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE

Major: Computer Engineering

Major Professor: Dr. Carolina Cruz-Neira

Iowa State University

Ames, Iowa

2000

Copyright © Lewis Charles Hill II, 2000. All Rights Reserved
ii

Graduate College
Iowa State University

This is to certify that the Master's thesis of

Lewis Charles Hill II

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy
DEDICATION

This thesis is dedicated to my niece Aleshia LaChelle. She is my personal reminder that the little people in life are the most precious (you go little girl).

This thesis is also dedicated to God, family, and friends. God makes all things possible, my family has helped raise me up to the height where I could achieve, and my friends have encouraged me onward when times got rough.

This thesis is also dedicated to the memories of our cherished friends and loved ones that have gone on.
TABLE OF CONTENTS

LIST OF FIGURES v

LIST OF TABLES vi

ABSTRACT vii

CHAPTER 1. INTRODUCTION 1
  Rationale 2
  Research Questions 4
  Scope of Research 5

CHAPTER 2. BACKGROUND 7
  Interaction 8
  Usability 20
  Intelligent User Interfaces 28
  Conclusion 36

CHAPTER 3. PALMTOP INTERACTION DEVICE IN IMMERSIVE ENVIRONMENTS 38
  Concept 38
  Design Details 50
  User Testing 66
  Conclusion 71

CHAPTER 4. RESULTS AND DISCUSSION 73
  Description of the Participants 73
  Test Round One 74
  Test Round Two 83
  Discussion 90

CHAPTER 5. CONCLUSIONS 91
  Palmtop Device Interaction Effectiveness 91
  Suggested Applications of Palmtop Interaction Device 93
  Future Research 94
  Final Summary 96

APPENDIX. USER TEST SURVEY FORMS 99

REFERENCES CITED 106
LIST OF FIGURES

Figure 2.1 Interaction Framework with System, User, Input, and Output 9
Figure 2.2 Interaction Framework and Translations between Components 10
Figure 3.1 JAIVE Users in the C2 53
Figure 3.2 JiMirror and JiInterface Block Diagram 57
Figure 3.3 Ji3dof Horizontal Slider Arrangement 60
Figure 3.4 Ji3dof Touch Panel Interface 60
Figure 3.5 JiOps 61
Figure 3.6 JiNote 61
Figure 3.7 JiInfo Text Viewer 62
Figure 3.8 JiInfo HTML Viewer 62
Figure 3.9 JiMap 63
Figure 3.10 JiColor 64
Figure 3.11 JiParam 64
Figure 3.12 JiPicture 65
Figure 3.13 Qbe User in the C2 68
Figure 3.14 Qbe Device 68
Figure 3.15 JAIVE/Builder and VADeT Shape Templates 68
Figure 3.16 Intermec Pen*Key 6642 Computer 70
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Shackel’s Usability Dimensions</td>
<td>25</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>Usability Assessment Methods</td>
<td>26</td>
</tr>
<tr>
<td>Table 2.3</td>
<td>Usability Defect Categories</td>
<td>27</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Test Round One Participant Responses to Questions 1-14</td>
<td>77</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Test Round One Participant Responses to Question 15</td>
<td>80</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Test Round Two Participant Responses to Questions 1-7</td>
<td>86</td>
</tr>
<tr>
<td>Table 4.4</td>
<td>Comparisons of First and Second Round Answers</td>
<td>87</td>
</tr>
</tbody>
</table>
ABSTRACT

Immersive Virtual Environments offer several new and exciting methods of interacting with computer systems. Virtual Reality systems provide users with a wealth of sensory information that can help us understand difficult visualization problems. The wealth of information and the variety of interaction tasks requires careful design of user interaction methods.

At present, interaction in Immersive Environments is often performed through functionally overloaded input devices or through custom 2D and 3D simulated interfaces. The simulated interfaces might include virtual windowing systems, simulated information displays or visual menus. In many cases the custom virtual interfaces are placed between the user and the virtual environment data, occluding the user’s view of the virtual world.

The placement of projection surfaces around the user in Immersive Projection Technology systems allows the integration of real world interaction devices. One device we want to use in an IPT system is a palmtop computer. Palmtop computers can facilitate interactions with large amounts of data, alphanumeric information, or abstract operations that do not map well to current VR interaction methods. This research effort discusses the potential uses of a palmtop interaction system to reduce function overload and present familiar 2D and 3D interfaces to the user.

Palmtop computing devices play an increasing role in our daily activities. Palmtop computing devices are portable, lightweight, easy to use, and present familiar Windows, Icons, Menus, and Pointer (WIMP) controls to users. A number of these devices feature high-resolution graphics, multimedia capabilities, wireless networking, and touch sensitive displays.

Palmtop computers can serve to augment our interactions in the virtual world. A palmtop interaction system can facilitate operation selection, parameter specification, object manipulation, vocal or text annotations, the display of online help and more. Further, the palmtop computer preserves the user’s sense of presence because it remains with the user’s person as part of the accessible environment. The palmtop
is accessible by the user before, during, and after the virtual experience.

Palmtop computing, VR technologies, and interaction methods have been combined in previous research efforts. The JAIVE project extends previous efforts by integrating wireless networking, utilizing Java™ cross platform user interface libraries, examining an increased number and variety of interaction operations, utilizing usability design methods, and accommodating custom designed interfaces. This effort combines Immersive Projection Technologies with the most recent improvements in Palmtop computing systems.

This thesis discusses the design and usability testing of a palmtop interaction system for projection based VR systems. The discussion will include an overview of our software system; the Java based Interface to the Virtual Environment (JAIVE). We will highlight the system's usability design considerations, which include, consistent operation, task organization, and customizability. We also address the use of Java™ technology to ensure cross platform appearance and operation, to accommodate the development and integration of new interaction types, and to provide compatibility with new palmtop computing devices. We also identify potential applications of the interaction device and identify future directions for the project including spatial awareness and adaptive user interface techniques.
CHAPTER ONE
INTRODUCTION

"Ok, what buttons do I push?" This is often the first question users ask when presented with a new immersive virtual environment. The next is usually "How do I stop this thing from spinning?" Add the comment "I can’t read that text" and the questions "How do I close this virtual menu" and "Where’s the undo button", and one can rest assured that a typical virtual reality interaction system has been designed. The typical interaction system is often difficult for the operator to understand.

Virtual Reality (VR) research aims to reduce the separation between the user and the computer. VR systems create artificial synthetic worlds that respond to user input [Burdea94]. VR applications can range from scientific to artistic and can be realistic or completely fictitious [VRAC00]. VR systems are available in many sizes, including wearable computers, head-mounted displays, stereo monitors, and multi-projector immersive rooms [Burdea94][Cruz-Neira93].

But beyond the technology, developing a good VR system requires the development of effective interaction techniques [Dix98][Burdea94][Hinckley97][Gould85]. How does one determine if an interaction technique is effective? It has been stated that the user should easily understand an interaction method [Norman90]. Just as people understand that stairs are made for climbing, computer users have notions of how computer systems should be used. Users have a large amount of experience with two-dimensional computer workstations that should be considered when an interaction system is designed [Norman90][Dix98].

While a computer workstation is familiar to the user, the keyboard and mouse from such a system are not designed for use in the virtual space. The mouse requires a flat surface; the keyboard must set on a surface and be placed at a comfortable angle for typing. The CPU housing and monitor are too bulky for the user to carry into a VR system. While it is not physically practical to carry a full sized desktop computer inside of an immersive virtual environment, recent inventions have provided us with other alternatives.
Palmtop computers have become increasingly powerful over the past years. Palmtop users can perform a number of tasks including, email, web browsing, word processing, stock transactions, schedule planning, and more. These devices present a familiar windows-like interface to the user and can be manipulated with a finger or small stylus. They provide a number of commercially available networking options and are of relatively inexpensive with prices ranging from $200 [Palm00] to $5000 [Qbe00].

Palmtop computers can make it possible to bring the power of the desktop computing system with us in our virtual environment interactions. The Palmtop's user interface allows the user to directly transfer understanding of computer interfaces to the virtual world. This thesis examines the suitability of palmtop devices for immersive VR interaction. This chapter discusses the rationale of the study, presents the research questions, and defines the scope of the project.

Rationale

Just as there are many VR systems in existence today, there are many interaction methods as well. These interaction methods are preformed through a variety of devices. Example devices include: wands, data gloves, pinch gloves, trackballs, SpaceOrbs, Phantom's and more [Hinckley97] [Dix98] [Cruz-Neira93] [Burdea94].

Each device has a number of different uses that are related to the physical characteristics of the interaction system. A wand is good for selecting and pointing. Data and pinch gloves are good if one needs to perform grabbing and manipulating operations [Perles98]. Trackballs can be used to rotate models or to scroll quickly through a list of options. SpaceOrb's can be used to navigate spaces or orient models [Labtec00]. Phantoms provide excellent force feedback and good accuracy over a limited range [SensAble00].

Interaction devices also may have limited accuracy due to the state of present day technology. Wands and gloves are suitable for object selection; however, fine manipulation may prove to be difficult due to tracker accuracy or stereo occlusion issues. Display resolution can also limit the effectiveness of an interaction method. Certain
sizes of text are not easily readable in the virtual environment [Ellis93]. Fine detailed work may not be possible if the visual quality is lacking.

An interaction method includes more than the physical design of the device. The interaction method also includes the relationships between manipulations of the device and some action in the virtual world [Dix98]. The user must understand both aspects to interact effectively. This mapping is often unclear due to poor communication of device operation to the user, nonstandard appearance of device controls, or input device function overloading [Norman90][Abowd91].

Functional overload means that the interaction device may perform a large number of functions within the same application. Users are often confused as to which operation they are performing at a given time. This confusion can lead to mistakes or incorrect decision-making. While the virtual environment might indicate that the user has performed an invalid operation, it is better to prevent mistakes by letting the user know which operations they are allowed to perform. There should be a clear understanding of the effects a user’s action will have on the system [Norman90].

Further, there are few standards in VR that define which types of interaction devices are used, how controls are drawn or placed, or what tasks are performed via interaction device. As a result, almost every application has different controls and interaction protocols. Every application must be learned or relearned by the user.

We know from human-computer interaction studies that learning is a cumulative process. Humans carry a tremendous amount of personal experience into computer environments. This experience can be used to learn new skills [Dix98]. Why should the virtual world feel like a completely alien environment to someone who works with computers constantly? Can we integrate some familiar interaction elements to help users learn new tasks? Is there a way to bring the familiar computer interface along with us in the virtual environment without overly burdening the user? These questions are of interest to us and have motivated the work presented here. To that end, we investigate technologies that could help us seek answers to the questions listed above. One example technology is that of palmtop computing devices.
Current trends in technology include the design of miniaturized palmtop computing devices. Palmtop computing devices have several characteristics that make them desirable for use in virtual environments. Palmtops present users with familiar windows, icons, and mouse pointer interfaces. Palmtops are lightweight, easy to use, and have high quality touch sensitive displays. Palmtops are designed to interface to larger computing systems using standard network connection technology [Palm00][PocketPC00].

It would be advantageous to bring the familiarity of these devices into the virtual environment. Equipping the user with palmtop computers mimics their real world activities. We believe that providing familiar devices reduces learning time, reduces user frustration, and increases productivity. This device could both instruct the user and receive input from the user. The device could be reconfigured as the application demands. The device is small enough that it does not restrict the users motion in the environment.

Research Questions

This thesis discusses the effectiveness of palmtop interaction devices in VR systems. We will examine interaction in two- and three-dimensional systems. We will discuss methods for evaluating the usability of an interaction system. We will identify which types of information can be transferred out of the virtual environment and into the palmtop device. Of this information, we must then identify which interaction tasks are possible.

We do not expect that the palmtop device will be the best method for every type of interaction. We will investigate which interaction methods can and cannot be improved with the aid of a palmtop device. More formally stated, we will address the questions: ‘Can the palmtop strengthen deficient interaction methods?’ and ‘Which types of interaction are most improved through the use of these techniques?’

Further studies will then be outlined in the area of device adaptability. In the long run we hope to determine if it is possible for this device to learn a user’s behavioral patterns. The work we
begin with project will lay the groundwork for such a study to take place.

**Scope of Research**

The field of VR interaction spans several disciplines including but not limited to psychology, cognitive science, usability, computer science, and ergonomics. Our discussion will not focus on every aspect of research from these areas. The research will focus on interaction in immersive and non-immersive systems using a palmtop device. It will discuss interaction methods and the usability characteristics of the methods. Background information will be presented regarding adaptive user interface methods to demonstrate the long term potential for research in this area beyond the scope of this thesis.

The prototype palmtop interaction system will be discussed in moderate detail. The discussion will center on the types of interactions implemented for this research effort. This discussion will clarify design decisions and identify any interface issues that may affect the outcome of the research. The software design issues will only be discussed in enough detail to illustrate relative design decisions.

Interaction device effectiveness will be presented as a combination of experimental data, previously conducted research, and personal experience with interactive systems. The thesis will describe a study of the prototype palmtop interaction device. The results of this study will be used to identify more effective and less effective palmtop interaction methods. Suggestions for the appropriate use of this device will be given and future directions for this research will be presented.

In summary, this research effort will accomplish the following:

- Survey background material in the areas of Interaction, Usability, and Intelligent User Interfaces
- Design an interaction model for a palmtop interaction device for immersive environments
- Integrate this model into a prototype palmtop interaction device
- Determine the effectiveness of the prototype system in immersive environment interactions through experimentation
- Identify future areas of research for this technology.
CHAPTER TWO
BACKGROUND

Our palmtop interaction system incorporates research from interaction and usability engineering. The future plans for the palmtop interaction device include adaptive and intelligent user interface techniques. Here we briefly describe the relevance of these research areas to the project and detail the information presented in this chapter.

In order to discuss effective virtual environment interaction for palmtop computers we must first survey existing methods of interaction with computer systems. The palmtop computer provides non-immersive interaction methods that will be combined with the immersive interaction methods of our VR systems. Therefore this chapter discusses both immersive and non-immersive interaction methods.

The palmtop interaction device's effectiveness will be measured with usability testing methods. Usability methods are applied to the design of the system from conception through completion. This chapter presents information about usability principles, design strategies, and evaluation methods.

An additional area of research associated with user interface development is that of intelligent and adaptive user interfaces. Intelligent user interfaces are designed to model, predict, and/or automate a user's tasks. These techniques can be applied to the palmtop interface design to assist the user's tasks. We will introduce the motivating factors surrounding the development of intelligent user interfaces. Intelligent user interface design criteria, capabilities, and effectiveness will be discussed. A number of example agents from will be presented as well. This information is presented to the reader to illustrate the long-term goals of this research effort.

To provide a brief overview, this chapter will discuss immersive and non-immersive interaction methods. It will introduce the usability principles we will use to design and determine the effectiveness of the interaction system. It will also introduce intelligent user interfaces as part of the future directions for the research effort.
Interaction

Introduction

What is Interaction?

Interaction in the context of human-computer interfaces includes any communication of intent or state between a computer user and a computer [Dix98]. An interaction system should provide the user with information about the state of the system or application. The system should allow users to perform system operations as necessary [Dix98] [Burdea94].

Foley, et al. identifies six main interaction tasks that a user may wish to perform [Foley84]. Watsen summarizes and describes these tasks [Watsen99]. Foley’s list, directly quoted from Watsen, includes:

- Selection: Make a selection from a number of alternatives.
- Position: Indicate a position on the display or in the workspace.
- Orientation: Alter the orientation of an object in the workspace.
- Path: Generate a path, which is a series of positions and orientations over time.
- Quantification: Specify a value (i.e. a number) to quantify a measure.
- Text: Input a text string.

Foley’s list of interactions is very action oriented. We turn to Donald Norman’s research to consider the thought processes involved in interaction. Norman’s Execution-Evaluation model of interaction, presented in [Dix98] and [Norman90], acknowledges two entities: the user and the system. Norman’s model has two main stages. The user forms a task and performs an action using the system. The user then evaluates the outcome of that action as presented by the system. Norman further divides the interaction model into seven stages:

- Establishing the goal
- Forming the intention
- Specifying the action sequence
- Executing the action
• Perceiving the system state
• Interpreting the system state
• Evaluating the system state with respect to the goals and intentions

These stages all focus strictly on the activity of the user. Norman’s model focuses on the system solely an interface [Dix98] [Norman90].

The next model we will examine considers the behavior of the system as well as the interface. Abowd and Beale [Abowd91] extend Norman’s model to incorporate any communications that the system sends to the user through the interface. In this model, the interaction system is divided into four major components: The User, the System, the Input and the Output. Operations form a cycle where the Input and Output serve as conduits between the System and the User. Figure 2.1, adapted from [Abowd91], indicates these relationships.

![Figure 2.1 Interaction Framework with System, User, Input, and Output](image-url)

Abowd and Beale’s model features four main operations: presentation, observation, articulation, and performance. Figure 2.2, adapted from [Abowd91], indicates these relationships.

*Presentation* is performed by the system, through the output channel, to indicate system status. The user *observes* the system output and formulates tasks. The user performs actions by *articulating* them to the system through the input channel. The system then *performs* the task that has been articulated through the input channel [Dix98]. The input/output channel plays an important role in both interaction models.
discussed earlier. Our next task will be to briefly identify input and output devices found in computer systems.

**Interaction Devices**

Computer system I/O devices provide a means for exchanging information between the system and the user. Typical output devices might include displays, speakers, and printers. Typical input devices include mouse, keyboard, joystick, and cyberglove. Some devices are both input and output devices. For example, force-feedback joysticks capture user actions and use haptic force or vibration to alert the user of some action in the environment. Some displays such as touch screens and palmtop devices are also input mechanisms [Dix98].

The physical characteristics of the input devices: size, weight, number, and placement of controls provide some hints as to how the device can and should be operated. One example is the classic Atari 2600® style joystick. Users generally understand that the device is supported with the left hand and maneuvered with the right. The button is placed so that the supporting hand can easily press it. Another example is tablet and stylus interaction device. The tablet is held like a note pad and the stylus is used as a pencil.

The device may also have a recommended style of use. The device may require a certain amount of space or range of motion for correct operation. A mouse is designed to roll over a flat area. This type of device would be very inefficient if constrained to roll on a very small area or if rolled over carpet. Another example is the SensAble Technologies Phantom™ haptic interface hardware [SensAble00]. This
device conveys the physical sensation of a virtual space to the user through either a stylus or finger grips which are attached to a mechanical arm. The length of the segments of the mechanical arm limits the device's range of motion.

Input/output devices also face technological limitations. These limitations often involve measurement accuracy and resolution [Hinckley97].

Computer systems of today can be grouped into two main categories based on the characteristic of immersion. Immersion is the user's sense of presence in the environment [Burdea94][Pausch97]. The next two sections will discuss interaction methods for both immersive and non-immersive computing environments.

**Interaction in Non Immersive Environments**

A non-immersive environment is one in which the user is not surrounded by the simulation they are interacting with. An example of such a system might include a desktop computer with a monoscopic monitor. The system input peripherals for such a system would typically include a keyboard, mouse, and possibly a gaming device of a sort.

**Non-Immersive Interaction Methods**

Users interact with these systems through common interface styles. These styles include [Dix98]:

- Command line interface
- Windows, icons, menus, and pointers (WIMP)
- Multiple degree-of-freedom input device interaction
- Vocal interaction

The command line interface provides access to system functions through a text interface. The command line interface prompts the user for commands and displays the outcome of the issued commands. This interface is found in many server workstations. A keyboard and mouse are often used to interface with the command shell. These systems allow the use of powerful customizable commands and wildcard operators [Dix98].
The WIMP interface provides a visual abstraction of operating system commands. WIMP was championed by the Apple Macintosh™ and has since been integrated into most other commercially available operating systems. Under WIMP, computer resources are represented as icons. Windows are used to display icons, access the file structure, and provide input to applications. Menus provide the user with a list of available operations. A mouse, track-pad, or track-ball device is used to perform most system operations. A keyboard may be used to enter shortcuts or accelerators for system operations. Some WIMP systems also integrate a command line interface to OS features as well [Dix98].

Multiple degree of freedom interaction devices can be with non-immersive computer systems to examine solid geometry or hyper-dimensional data sets from various perspectives. These devices can be used in combination with a mouse and/or keyboard to modify the datasets. Computer Aided Design systems often feature this style of interaction. In these applications, the designer positions the model with a 3D interface device in one hand and selects points with a 2D selection device in the other [Hinckley97][Dix98][SensAble00].

Voice interaction methods allow a user to verbally issue computer requests. The system interprets the vocalization and performs the most appropriate task. These systems may operate by recognizing key commands from a user, or may perform a natural language analysis on the spoken phrase. Some heralds these systems as the future of computing for their ease of use, hands free operation, and quick learning. However, there are a number of potential disadvantages to this type of interaction. Speech recognition systems often require training to understand users' spoken phrases and some systems cannot discern heavy accents. Speech recognition applications are sensitive to background noise. Inadvertent command execution is a possibility and may have serious repercussions [Burdea94][Cai96][Vo93].

**Effectiveness of Non-Immersive Interaction**

Non-immersive interaction methods have been in use for a long time and they are fairly mature. However, they still have some limitations. As we have seen with command line interfaces, users must remember potentially cryptic system commands to accomplish tasks.
Regardless, command line interfaces are very popular. Masterful users combine operations, command line arguments, wild card operators, and customized scripting languages to perform complex operations.

WIMP systems require a moderate amount of hand-eye coordination to be used successfully. Users may not immediately grasp the visual representation of WIMP system resources. Command line users express a lack of control when operating WIMP systems citing that the interfaces are slow or too restrictive. Today's WIMP systems have become increasingly flexible by combining WIMP methods with command line interfaces. These systems allow users to operate in a variety of modes, as they feel necessary.

Hand-eye coordination is also an issue with multiple degree-of-freedom interaction methods. Multiple degree-of-freedom devices are very intuitive to users because they reflect natural object in hand manipulation methods. Stationary multiple degree-of-freedom input devices can suffer from limited ranges, inaccuracy, or over sensitivity. In some cases, the user can manipulate the data, but not as skillfully as if they were maneuvering a real object. Many of the shortcomings in multiple degree-of-freedom devices have been compensated through improved software design. These devices are used heavily in computer aided design, rotoscopic model digitization, and computer game systems.

There are also limitations whenever high dimensional data is interacted with in non-immersive systems. The lack of immersion can limit the user's perception while interacting. It can be difficult in non-immersive systems to determine the size, depth, and orientation of onscreen objects. Most non-immersive systems require tedious manipulation of input devices with one or both hands. Users are not free to use their bodies while interacting. At best the user has real-time control of a representation of a person performing a task.

Interaction in Immersive Environments

Immersive environments aim to envelop the user and inspire a sense of presence in the virtual world [Pausch97]. The user is surrounded by the imagery. As my young friend D.J. Myers II, a third
grader from Ames, IA, mentioned "The images jump out at you!" This quality of immersion combined with interaction and imagination form Burdea's definition of a Virtual Reality system [Burdea94].

**Immersive Environment Systems**

Virtual Reality systems exist on several scales, which include:

- Stereoscopic Monitors
- Stereoscopic Powerwalls and Panoramas
- Head Mounted Displays
- Surround Screen Immersive Environments

VR Systems feature six degree of freedom user tracking, stereo imagery, and high quality displays [Cruz-Neira93]. A large number of interaction devices can be used in VR systems. These devices include:

- Wands or similar pointing device with buttons
- Gloves
- Full Body Motion Capture Suits
- Trackballs
- Three Dimensional Probe and Rotoscopy Tools
- Force Feedback Mechanisms
- Driving or Flight Simulation Rigs
- Voice Interaction

These devices are used to perform a number of interactions. Common interactions for VR applications include object selection, object manipulation, gesturing, user interface window operation, and voice interaction. There are also a number of application specific interaction methods. We now describe these interaction methods in further detail.

**Immersive Interaction Methods**

Object selection is often performed using a tracked input device like a wand or a glove or with a number of three-dimensional pointing devices. Selection may be based on the pointing device's proximity to a virtual object, by ray casting from the pointing device location, or through volume-based selection. If the user is wearing a pinch or data glove, pinching the corner of or grabbing the center of an object may
indicate a selection operation. Wand users often press or depress and hold a button to indicate selection.

Ray casting can be used to select objects at a further distance from the user. The user aims the input device toward the virtual object. A virtual ray is extended from the front of the input device. The object that is closest to the ray is considered selected. There are variations on this method [Poupyrev96].

Volume selection can be used to pick a group of virtual objects. This selection can be performed with a bounding-box metaphor, or with dual input devices. A bounding box is the three-dimensional equivalent of a two-dimensional selection window. In the virtual environment the user might indicate opposite corners of a box or spherical selection region with a wand or glove device [Chan99][Hill99][Hill99b]. Dual input devices could be used to simultaneously select two corners of a bounding volume. A separate action is performed to complete the selection process. An example of a commercial product featuring two-handed interaction is Multigen Corporation’s SmartScene [SmartScene00].

Object manipulation includes the transformation or deformation of virtual objects. Object transformations might include translation, rotation, scaling, or shearing. These operations are typically performed with wand, glove, trackball, or three-dimensional pointing device. A glove-based system might translate or rotate an object if the user has grabbed it with one’s hand, mimicking the behavior of the object in the real world [Chan99][Hill99][Hill99b]. The same system might scale or shear objects if the user grabs and moves opposing corners of the object. Deformation can be accomplished by having the user transform an object with one hand and shape the object with the opposing hand [Perles98].

Navigation in the virtual environment is the process that the user performs to change position in the virtual space. Many VR systems allow the user to physically move around within the effective range of the tracking system or confines of the immersive projection technology system. Other VR applications require the user to navigate, effectively moving the center of virtual reality system to a new portion of the model. Navigation consists of several operations two of which include wayfinding and locomotion. Wayfinding is deciding the path one wishes
to travel along. Locomotion is the process of selecting heading and speed for traversal of the environment [Bowman97][Darken98].

Gesture based systems interpret user motions or actions as operations in the virtual world. Some gesture systems use sign language as an input mechanism. Others monitor the position of a user’s body. Gesture input systems are often implemented with a data glove or full body motion capture suit [SmartScene00].

There have been a number of efforts to incorporate WIMP-style user interface windows into the virtual environment. Many of these efforts have aimed to create virtual windows. The Virtual User Interface [Heath98], Virtual Windowing System [VRAC00], and the Brown University Interactive 3D Interface Toolkit [Zeleznick93] provide interaction windows with sliders, buttons, text input areas, and/or scroll bar mechanisms for presenting information to users.

Speech recognition systems for VR applications operate similarly as discussed above. The virtual environment provides many opportunities for the use of natural language processing. There are a number of artificial intelligence applications that use speech recognition to communicate with virtual agents [Rickel97][Vere90]. Speech recognition systems can also be used to communicate with other participants in multi-user distributed applications. Similar restrictions apply to both immersive and non-immersive voice applications as listed earlier. An additional problem arises when there are multiple participants in the same VR system. It becomes difficult for the system to distinguish a single voice from the group of participants.

Customized interaction systems are developed for applications when special input/output devices are required. Flight simulation, virtual prototyping, and human in the loop simulation bring their own set of challenges and interaction methods to the virtual environment. Flight simulators and virtual prototyping applications often utilize custom interface rigs. One example is the Deere and Company Tractor research project at the ISU Virtual Reality Applications Center [VRAC00]. This project uses a tractor seat buck that has been instrumented to enter physical control information into the virtual vehicle simulation. The virtual environment is used to simulate the appearance and operation of the exterior portions of the vehicle.
Designers and operators can work simultaneously in the virtual environment to evaluate product visibility, ergonomics, and operability.

**Effectiveness of Immersive Interactions**

Virtual Reality systems are powerful tools that may be used in a wide range of application domains. The current state of technology imposes limitations on interaction effectiveness. These limitations can stem from technological, visual, ergonomic, or device design issues.

Technological issues in immersive virtual environment systems include limitations in visual clarity, field of view, and scene complexity. We begin the discussion on visual clarity by noting that VR systems vary in display resolution and size. Head mounted displays feature high quality displays over a limited field of view. Immersive Projection Technology systems like the C2 and C6 [VRAC00] have high-resolution graphics spread over a large field of view. Display resolution can affect the visual clarity of the objects in the scene. The human visual system's acuity determines the level of detail that we are capable of perceiving. Discussions of human visual perception for VR systems can be found in [Ellis93][Burdea94].

VR systems often sacrifice detail to maintain high frame rates. Scene complexity must be reduced to allow interactive operations. The combination of these factors can in some cases lead to unreadable text and oversimplified virtual models.

Physical input devices in projection based interaction systems always appear in front of the virtual objects. This condition is called stereo violation. When stereo violation occurs it is difficult to determine the depth relationship between the physical device and the virtual object. Stereo violation hinders object selection and object manipulation operations.

Ergonomic issues with VR input devices can include poor design, excessive weight, unwieldy size, or poor design. Users can become fatigued or perform inaccurately if the device is too heavy or requires too much effort to move. User's accuracy can also be limited if the size of the device causes excessive stereo-violation. Improper input device control design can cause operational difficulties for the user.
The input device can also restrict the rate at which data can be entered into the system. VR users do not typically use a full-sized alphanumeric keyboard for entering data into applications. A mini-keyboard mounted on one arm, while functional, only allows one hand to type at a time. There are one-handed alphanumeric chording devices. These devices can input a large vocabulary of commands through a five-finger interface. These devices require a lot of user effort to master and the maximum input rate is lower than that of QWERTY, Alphabetic, and DVORAK keyboard layouts [Dix98].

The input device may also have inaccurate measuring mechanisms. Magnetic, acoustical, visual, and physical tracker data are all susceptible to error and interference from implicit design issues, and outside interference. Input device sampling resolution might be too small to provide smooth precise data. Slow update rates may miss user input actions or, conversely, fast update rates may resample the same user action.

Problems with Immersive Environment Interaction Metaphors

There are a number of unresolved issues with interactions in the immersive environment. We will discuss these issues using the terms provided in the Abowd and Beale model for interaction. We shall focus on the presentation and observation of the output and the articulation and performance of the input.

The presentation to the output concerns the system's representation of the virtual environment to the output channel. There are no widespread API's for immersive VR application interface development. Each virtual windowing system has a different appearance, each menu a different layout, and each control a different use. Users are required to learn new interaction methods for each and every application they encounter. This does not leverage user experience and can lead to several incorrect operations be the user.

When the user observes the output, the individual may not comprehend the visual representation of the virtual action. The action that the user performed may not seem to match the new images or the new presentation of system state. The non-uniform appearance of virtual controls can cause ambiguity as to what operation has been performed,
or which operations are available for selection. One example output
comprehension problem might involve the lack of gravity in a virtual
architecture application. While it is beneficial to have a gravity free
environment for three-dimensional model exploration, it can be
disorienting for an architect's design to float in virtual space.

Articulating one's wishes to the system is often a problem in the
virtual environment. How do the input devices work? What operations
does the input device perform? We have established that many interfaces
are unique in appearance and operation. Input devices and virtual
controls may operate differently than users anticipate either from
their experiences with other applications or from the real world.
Researchers may be hesitant to label virtual controls for fear of
losing realism or sacrificing visual quality.

Another articulation issue is precision. An architect may decide
to place a beam at precisely 5.24531 meters along some axis.
Electromagnetic tracking system limitations and human inaccuracy would
make this task nearly impossible with wand or glove based manipulations
in a one to one full-scale design. However, alphanumeric input, if
available, could fix the problem easily.

Finally we must consider the system performance of the input
operations. Again, the users expectation of what an operation should
do comes into play. Dropping a real object means it falls to the
ground. This may not be the case in the virtual environment. Newtonian
physics applies to real objects, but these rules may be suspended or
modified in the VR world. Even if we correctly articulate our
intentions to the system, the system may be designed to interpret the
actions differently that we would expect. Again this can lead to user
confusion.

We now conclude our discussion on interaction: This section has
presented two models for interaction, discussed interaction methods for
immersive and non-immersive environment. A number of limitations were
presented for each type of interaction device. The next section will
examine the principles of usability of objects and interfaces.
Usability

Introduction

Designing a product for good usability means that in the end, users can effectively and efficiently use the product to perform tasks [Hackos97]. The field of usability provides guidelines that encourage good design principles from the initial concept through the final stages of a design. Usability methods are particularly helpful when applied to the design of user interfaces. The palmtop interaction device integrates these usability methods to strengthen interaction metaphors. This section will present usability principles, design strategies, evaluation methods, and will discuss common misconceptions regarding usability in the design process.

Usability Principles

Gould and Lewis present key principles for usability. Their recommendations include three main principles for design: early focus on users and tasks, empirical measurement, and iterative design [Gould85].

Focusing on the users and their tasks from the beginning of the design process encourages developers to understand how the product may be used. The designer and the user may have dramatically different use patterns with a product or system even if they have the same goals in mind [Norman90]. Failure to recognize this difference until late in the design process may mean reduced product usability or could lead to a costly redesign. To prevent this problem, Gould and Lewis suggest that developers study the users' attitudes and behaviors. They also advocate understanding the users' thought processes and physical characteristics [Gould85].

Empirical measurement of user performance is also critically important early in the development process. Empirical measurement provides a means to judge improvements in the design over time [Gould85][Nielsen93a]. It is important that empirical measurement is performed with adequate metrics. It is also important to consider the costs associated with testing and the accuracy of the measurements.
Methods for performing empirical measurement are presented later in the thesis.

The third principle presented by Gould and Lewis is to refine the design iteratively. Whenever a design flaw is discovered, the product must be fixed and retested [Gould85]. In developing a usable product, it is quite possible that what the designer perceives to be an improvement may be a hindrance to the user [Norman90]. Iterating the design and test processes should help to reveal these problems as they arise [Gould85].

Nielsen [Nielsen93b] presents a similar definition of usability. Nielsen identifies five attributes, which include learnability, efficiency, memorability, errors, and satisfaction. Learnability refers to the user's ability to learn the system. Efficiency means that the user should be able to efficiently operate the system in order to maximize productivity. Memorability means that users should be able to memorize and easily recall how to operate the system. The system should maintain a low error rate and facilitate error recovery. Finally, Nielsen states that the user should enjoy using the system [Nielsen93b].

**Design Strategies**

**The Design of Every Day Things**

The best time in a design cycle to consider usability is in the design process is all of the time [Gould85]. In the previous section we reviewed a number of perspectives on what usability is. Now we will examine strategies for designing usability into systems. Donald Norman's design strategies presented in "The Design of Every Day Things" include seven concepts ranging from cognitive modeling to error compensation. The ideas Norman presents are [Norman90]

- Use Existing Knowledge
- Simplify Tasks
- Increase Visibility
- Present Correct Mappings
- Exploit Constraints
- Designing for Error
- Standardize Controls and Operations
The first of Norman's design guidelines is to use existing knowledge. As mentioned in the introduction, users have a wealth of experience in the real world. This information can help or hinder the user when learning new tasks. Donald Norman explains the role of user knowledge in his first usability design principle: use existing knowledge. In describing existing knowledge, Norman acknowledges three mental models in a system. These are: the user's model, the design model, and the system image. The user model is everything relating to how the user "explain[s] the operation of the system" [Norman90]. The user's previous knowledge and experiences affects the user model and must be considered by the designer. The design model is the designer's concept the system and it's operation. The system image is the actual appearance and operation of the system. Norman emphasizes that it is important for the system image to reflect all of the information a user requires to perform a task [Norman90]. It is important to note that Norman's concept of a user model is different from user modeling techniques used in the design of intelligent user interfaces discussed in the section of this chapter titled Intelligent User Interfaces.

Norman encourages designers to simplify tasks. This can be done by a combination of keeping the task the same throughout the system, automating the task, changing the task, or increasing the visibility of the task. Increasing visibility can include making normally invisible background tasks visible to the user. Keeping the user aware of the status of a task improves feedback and enables the user to maintain control of the system. Norman states that increased visibility can help bridge the gap between the user's execution of a task and the user's evaluation of the outcome of that task [Norman90].

Norman also emphasizes the importance of correct mappings between the user's intentions and actions that are available to the user. The user should also understand the relationship between one's actions and the effects of those actions on the system. Norman suggests consistent mappings between the actual system state and the user's perceptions of the system [Norman90]. Virtual environments present visual, aural, and textural information to immerse the user [Burdea94]. All of this information must consistently and accurately present the system state.
Further, the user’s perceived system state must map to the needs, intentions, and expectations of the user [Norman90].

Design constraints can be found in a number of real world devices. For example, light switches have distinct on and off positions. Car steering wheels transmit steering force information to the user’s hand proportional to the force of turning the vehicle. Norman advocates that natural and artificial constraints should be exploited by the system [Norman90]. One example user interface constraint can be found in a scroll bar for a user interface window. While this is obviously an artificial constraint, it indicates the end of a document to a user.

A system designer should expect users to make errors [Norman90]. While it is very difficult to design for every mistake a user might make, early focus on the users and their tasks may reveal common sources for error [Gould85]. The system should support users’ intentions instead of fighting them [Norman90] and adequate responses should be given to incorrect operations [Dix98].

Norman’s final comment is to standardize. Standards and conventions may not be the optimal design, however consistency between system designs will aid users in learning the new task [Norman90].

Usability Misconceptions

Norman, Gould, Nielson and many others have presented and demonstrated methods for developing usability. Discussions still persist about how to demonstrate the benefits of usability to designers, companies [Bloomer98], or the computer industry [Maguire93]. There are several sources of misconceptions of usability principles or their effectiveness in design [Gould85]. This section will present issues surrounding the misinterpretation and undervaluation of the usability principles regarding early focus on users, empirical measurement and iterative testing. This section will also distinguish between apparent and inherent usability [Kurosu95].

Gould cites the misinterpretation of usability principles as a detriment to the effectiveness the design process. The usability principle of early focus on the user is often misinterpreted to mean
identify the target user group rather than understand the user. The end user's input should be considered early in the process instead of after a prototype system has been designed and implemented. Another reality in product development indicated by Gould is that design decisions made in the prototype of a system are often incorporated into the final product. This prototype may not have been intended to be the final product, but project or resources may have forced the prototype to become the final product. Gould and Lewis also state that it may be difficult to develop a solid understanding of users for a new product if there is no sufficiently similar system to compare against [Gould85].

The usability principle that encourages empirical measurement of user performance is often misinterpreted to mean perform a system functionality test. Gould and Lewis state product developers think differently about the products use and internal operation than users. Further, expert users do not operate the system or make the same mistakes as novice users [Gould85]. System test may not provide accurate usability information because these tests often involve demonstrating the product to customers. It is implied that the contractual relationship between the customer and the developer may not allow accurate feedback to be given [Gould85]. Empirical data also helps usability designers justify decisions. This data may be helpful in the presence of subjective views offered by influential members of the designing organization [Nielsen93b].

Iterative testing is rarely performed with sufficient vigor. A well-documented process needs to be created to track modifications to the system over time [Gould85]. Testing needs to be performed often enough to ensure reliable and valid data. In other words, the testing should ensure that we have repeatable results and that we have actually tested the usability criteria we intended [Nielsen93b].

Gould and Lewis continue to discuss why they feel that usability principles are undervalued. The main sources are: devaluation of the principles, confusion or competition with similar but different design ideas, underestimating the value of interaction with the end user group, and the erroneous belief that the methods are impractical [Gould85].
Usability Evaluation

Usability can be measured along a number of dimensions as we have seen above. These dimensions are not mutually exclusive. However most usability dimensions acknowledge the separation between the user's performance with a system and the user's attitude regarding the system [Lindgaard94]. Here we further examine dimensions for usability measurement.

Shackel [in Lindgaard94] lists four main dimensions for usability evaluation. These dimensions include: Effectiveness, Flexibility, Learnability, and Attitude. Table 2.1 describes these effectiveness dimensions [Lindgaard94].

Table 2.1 Shackel's Usability Dimensions

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectiveness</td>
<td>The user's performance measured in speed or accuracy.</td>
</tr>
<tr>
<td></td>
<td>The proportion of a user population that achieves a performance measure.</td>
</tr>
<tr>
<td>Flexibility</td>
<td>The amount of variation in task completion methods available to the user.</td>
</tr>
<tr>
<td></td>
<td>A system with too much flexibility is complex.</td>
</tr>
<tr>
<td>Learnability</td>
<td>A new user's ability to accomplish tasks.</td>
</tr>
<tr>
<td></td>
<td>The amount of training to obtain some level of effectives.</td>
</tr>
<tr>
<td></td>
<td>The number of times users request instruction before completing a task.</td>
</tr>
<tr>
<td></td>
<td>The time an intermittent user requires to relearn system features.</td>
</tr>
<tr>
<td>Attitude</td>
<td>The user's acceptance of the system.</td>
</tr>
<tr>
<td></td>
<td>A measure of user effort, stress, or frustration involved in interacting with the system.</td>
</tr>
</tbody>
</table>

Several methods have been developed to perform empirical measurement of these dimensions. These methods vary from attitude questionnaires to user observation with checklists. We now introduce a number of usability measurement and analysis methods.

It is important to develop a profile of the user group. This profile will provide means for comparing user's performances with the system. Background information should be gathered about any user trait
relevant to the task to be performed. [Rubin94] presents typical characteristics that are gathered for software usability testing. Rubin suggests that the following information should be gathered: demographics, computer experience, application or product experience, and user preferences [Rubin94].

The next stage in testing is to gather empirical data. Empirical usability data measurement methods along with an abbreviated list of positive and negative aspects of the methods are listed in Table 2.2 [Rubin94][Nielsen93b].

**Table 2.2 Usability Assessment Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Main Advantage</th>
<th>Main Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interviews</td>
<td>• Flexible</td>
<td>• Time</td>
</tr>
<tr>
<td>Observation</td>
<td>• Allows observation of real users with the system</td>
<td>• Appointments difficult to schedule</td>
</tr>
<tr>
<td>Performance</td>
<td>• Easy to compute results</td>
<td>• May not reveal specific problems</td>
</tr>
<tr>
<td>Measures</td>
<td>• Determine users’ subjective preferences</td>
<td>• May not be formed correctly or questions could be misunderstood</td>
</tr>
<tr>
<td>Questionnaires</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thinking Aloud</td>
<td>• Pinpoints users’ misconceptions</td>
<td>• Not natural for users</td>
</tr>
</tbody>
</table>

Selecting an appropriate usability assessment method depends on the system being evaluated. The system characteristics, system complexity, cost of testing the system, number of available users and effectiveness of the individual heuristics drive the choice of assessment method. Doubleday and Nielsen have performed comparison studies on a number of the usability metrics listed above. Doubleday’s experiments evaluated the usability of a novel information retrieval system [Doubleday97]. Nielsen compares heuristic, formal, empirical testing methods and presents a detailed cost-benefit analysis [Nielsen93a].

Once the assessment method has been selected the next task is to identify what aspects of the user’s experience we wish to capture. Maguire presents a detailed list of potential objective measurement criteria. Usability studies might consider the user’s performance, attitude, or mental effort and stress. Another approach is to perform
formal analysis of the system to assess the operational complexity of using it [Maguire93]. Nielsen also presents a set of heuristics in [Nielsen93b].

After experiments have been completed, statistical analysis must be performed to analyze the data. The actual data analysis methods are beyond the scope of this chapter. More relevant to this discussion is the characterization of typical usability experiment observations. User interface usability problems usually involve navigation, screen design and layout, terminology, feedback, consistency, modality, redundancy, user control, or an inconsistent match with defined tasks [Lindgaard94]. Table 2.3 better explains the defect categories with emphasis on the factors relevant to this research effort [Lindgaard94].

This section has introduced the concepts of usability and usability test methods. Expert and experimental information has been

<table>
<thead>
<tr>
<th>Table 2.3 Usability Defect Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
</tr>
<tr>
<td>Navigation</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Screen Design and Layout</td>
</tr>
<tr>
<td>Terminology</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Feedback</td>
</tr>
<tr>
<td>Consistency</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Modality</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Redundancies</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>User Control</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Match with User Tasks</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
presented to support the effectiveness of usability design methods. Usability misconceptions were also addressed. Usability design methods can be applied to the development of the palmtop interaction device. Further, the knowledge gained from user interface usability studies transfers well due to the 2D nature of the palmtop interface. The listing of typical usability defect categories provides guidelines for specific areas of focus while developing this system. This concludes the discussion on usability. In the next section we introduce intelligent user interfaces as help agents to increase the usability of the palmtop interaction system.

**Intelligent User Interfaces**

The next topic for discussion is intelligent user interfaces (IUI). Intelligent user interfaces attempt to enhance a user's ability to perform tasks. This type of interface allows the system to change its output presentation or alter its performance of the user's inputs. Developing an intelligent user interface for the palmtop interaction device could allow the system to simplify and/or automate user tasks in the virtual world. The remainder of this chapter discusses a number of artificial intelligence agents, presents criteria for agents, and discusses the role of the agent in applications. We will also discuss factors that impact the effectiveness of the adaptive interface.

**Motivations for IUI**

Intelligent user interface development is motivated by a combination of several factors, one of which is product complexity. Quinn and Russell noted at the Computer Human Interaction conference in 1986 that increased technology presence in today's products has provided us with increased functionality. Some consumers view this additional functionality as increased or overwhelming complexity. Overly complex systems can discourage customers from buying the product, frustrate users to the point of dissatisfaction with the product, or result in the under use of the product's features [Quinn86]. In Quinn and Russell's early view, intelligent interfaces
were meant to recognize the user's goals, help the user understand the procedure for accomplishing that goal, to observe the user's progress with the procedure, and to help the user perform and remember the procedure [Quinn86].

With the advent of the Internet, high-powered multimedia computers, and ubiquitous computing, intelligent interfaces began to play an expanded role. Object oriented software development and user interface API's have allowed the user interface to be developed separate from the application and be less dependant on the particular hardware system. The user interface remains separate from the underlying software structure and functionality of the application.

The increased flexibility of user interfaces systems has promoted diversity in the structure and functionality of the intelligent user interface. One function the intelligent user interface may perform might be that of a helper agent. Rickel and Johnson, from the Information Sciences Institute at University of Southern California, state that the interface should help the user by providing helpful suggestions [Rickel97]. In the case of Internet browsing, an intelligent user interface might try to simplify or accelerate the retrieval of information [Wasfi99]. The system could automate or simplify data representation and multimedia display tasks [Stephanidis97][Birnbaum97]. There is also the possibility of creating simulated assistants [Miller99]. These assistants add a lifelike face to the interface and in the case of VR systems; they can become intelligent instructors for the virtual world [Rickel97]. Before discussing these applications in greater detail, we will establish a set of criteria that can be used to describe intelligent user interfaces.

IUI Criteria

Intelligent interface concepts can be applied to a variety of platforms from palm-top computers to helicopters and from high-end Immersive Virtual Reality systems to VCR's. While there is no complete consensus as to the characteristics, behavior, or essential components of these systems [Stephanidis97], there are a number of well-accepted
criteria. IUI systems are typically adaptive, autonomous, collaborative, and robust [Brown99]. Collectively the software components that embody these characteristics are referred to as agents. We discuss agent adaptation, collaboration, and robustness below.

Adaptivity refers to the agent's ability to sense and adjust to the environment and the user. Brown adds that the purpose of adaptation is to "determine user intent and [improve] assistance" [Brown99]. Autonomy is "the ability to sense, act, and react... without direct intervention" [Brown99]. Adaptivity and autonomy combine to describe the agent's decision-making strategies. Detailed methods for decision-making in IUI interfaces can be found in [Stephanidis97].

Intelligent user interface agent collaborative behavior can include making suggestions, performing helpful behavior, or communicating with other software agents. Brown also indicates that part of the intelligent agents behavior is to predict the helpfulness of potential suggestions. The agent must evaluate the correctness and appropriateness of its actions [Brown99], correct its model of the user, and adjust its future predictions [Brown99][Birnbaum97][Wasfi99].

Robustness refers to the agent's ability to operate with unexpected or incorrect user input and to extend itself to new tasks. Robustness also includes the maintainability of the system and the capability of "gracefully degraded performance" if sensory information becomes unavailable [Brown99].

This section has listed general IUI criteria. These criteria alone do not develop a sense for how an agent might facilitate palmtop interaction. The next topic for discussion is to examine the role of the agent in various systems.

Role of IUI Agent

The Human Computer Interaction (HCI) and Artificial Intelligence (AI) communities have differing views as to the nature of the intelligent user interface. One perspective from HCI is that the agent is a tool for interaction. Under this view the agent automates processes and is primarily an input mechanism for the user [Brown99].
The AI community view is that the agent is an assistant [Brown99]. This agent can assist with instruction [Brown99] and perform active user task monitoring [Birnbaum97]. In this view the assistant is more likely to make suggestions than to perform actions on the system [Birnbaum97][Ericsson98]. The agent also operates in real time, adjusting to, and assisting the user when the need arises. The assisting agent also studies the user's actions to gather any information available. Methods for learning user behavior and creating user profiles are discussed by [Wasfi99]. The next type of interface combines together both the HCI and AI views of intelligent interface agents.

A mixed initiative user interface shares characteristics of both a tool and assistant. This type of interface offers intelligent services and user collaboration. The interface determines its behavior based on the confidence of its predictions about the user [Horvitz98]. This interface often offers multiple interaction modalities. An example would be a system that allows the user to input data through voice, keyboard, or touch screen device, at their own discretion. Studies have indicated that multiple interaction modalities can facilitate improved error handling [Oviatt99]. In these systems, users tend to select the least error prone interface method based on the task to be performed. Multi-modal interfaces with speech recognition can reduce the complexity of natural language processing because the "user's language is simplified" [Oviatt99]. This in turn reduces recognition errors. Oviatt indicates that if recognition errors occur in multi-modal systems, users tend to select a different interaction method. Oviatt also reports less subjective user frustration with the errors that do occur [Oviatt99].

**Effectiveness Issues**

In the last section, different views of IUI's were presented. In this section we overview design issues that can have an impact on the performance, accuracy, acceptance, and/or usability of these systems. One issue is execution time. IUI's typically execute search algorithms to perform predictions and decide actions. The execution
time of the AI algorithm affects system latency. The interface must balance AI algorithm complexity and execution time to ensure a responsive system [Birnbaum97]. The correctness and execution time of the search algorithms can be affected by the modeling of the user’s behavior. Poor search heuristics can further reduce the AI performance. Mistakes in AI computation can be expensive to verify and correct [Birnbaum97].

A second performance issue also pertains to user modeling. The purpose of user modeling is to facilitate predictions of the user’s behavior. All of the system predictions are based on the user model and any inaccuracies in the modeling are reflected in incorrect, inappropriate, or sub-optimal agent decisions. The IUI’s representation of the user can be improved through the use of several techniques.

Hornoff presents one such technique for a UI menu system. Hornoff presents research stating that humans select numerically or alphabetically ordered menu items faster than unordered items. The system’s cognitive user model incorporates that characteristic to improve user performance. The research continues to model the human menu item selection process using the EPIC (Executive Process-Interactive Control) cognitive architecture [Hornof99].

Another technique to improve user modeling involves the use of multiple data sources. The user model accuracy is limited by the relatively narrow communication bandwidth between the user and the interface. Quinn and Russell’s early research on user behavior detection addresses this limitation. They state that “the inability of the machine to detect nuances of user behavior...” degrades user prediction performance [Quinn86]. It is well accepted that human communication extends beyond the verbal content of a message. Our use of vocal tone and gestures provides a wealth of emotional content that computer systems cannot detect through text or menu based systems. This has two implications for improving user-modeling accuracy. The first implication is that more input of various types is needed from the human subject [Oviatt99]. Second, the input needs to be interpreted effectively by the system to prevent errors and to ferret out nuances in behavior [Quinn86].
Other interesting effectiveness issues noted by researchers is the apparent contradiction between Intelligent User Interface and Usability design principles [Birnbaum97]. An interface that reconfigures itself autonomously is unpredictable [Höök99]. If the interface performs task automation without explanation it violates the user's understanding of the system. This may leave the user sensing a lack of true control over the system. Höök presents methods for performing automation and interface reconfiguration while minimizing the impact on transparency, user control, and predictability [Höök99].

Another usability difficulty for IUI systems is communicating the AI decisions to the user. If the user never understands why the interface has changed, there is a gap in the user's understanding of the system. This problem is further complicated because AI search algorithm outcomes are difficult to represent in a clear and concise form [Birnbaum97]. Even if represented well, presenting the outcome to the user may interrupt the user's thought processes [Birnbaum97].

The effectiveness of an intelligent user interface agent can be limited by unrealistic expectations on the behalf of the developer(s). One common misconception is that the use of an IUI precludes the need for good interface design principles [Höök99][Birnbaum97]. The intelligent interface cannot be expected to rectify a poor interface design. Höök states that it is important to "distinguish the adaptive features of the system from the general usability of the designed tool" [Höök99]. Alternatively, the system's adaptive features can indicate which parts of the system cause the user the most distress. Steve Roth from Carnegie Mellon University's Robotics Institute notes that for one IUI project, integrating better interaction techniques sometimes eliminated the need for AI techniques [Birnbaum97].

Just as usability is often relegated to the last stage of a project, many early IUI applications are designed as late additions to interfaces. Lieberman from MIT media labs advocates a more integral role for IUI agents [Lieberman99]. However, if every interface becomes tailored for a specific user's preferences, what will happen to the interoperability of computer systems? How does one document a system that constantly changes its behavior? How will product design be affected long-term? These questions remain open.
To summarize, slow performance, inaccurate or incomplete user modeling, or improper AI predictions can impair the effectiveness of an intelligent user interface. User modeling and prediction can be improved by incorporating additional sources of user information, improving behavior nuance detection, or including information about human behavioral characteristics. The contradictory goals of usability and intelligent user interfaces illustrate the necessity for thoughtful and thorough integration of the systems and the applications. We now examine a number of existing intelligent user interface applications.

Example Agents

In the previous sections we introduced a number of issues pertinent to IUI design. This section will present a number of example IUI systems. Systems will be categorized by functionality as associate systems, information retrieval assistants, and data presentation systems. All of the applications include some level of the characteristics presented in the section entitled IUI Criteria.

Associate systems, as defined by Miller, are characterized as those which [Miller99]:

- Exhibit behavior of a capable human
- Perform the same activities of a comparable human expert
- Take initiative to perform tasks
- Formulate actions based upon the user's long term behavior

One example of an associate system is the US Army's Rotorcraft Pilot's Associate (RPA) [Miller99]. This associate assists helicopter pilots in performing in-flight tasks. RPA is capable of intelligent information management. The system is sensitive to the pilot's tasks and dynamically configures cockpit displays as is appropriate [Miller99].

Associate systems can be very powerful when used in virtual environments. These agents can be outfitted with a human representation. Some example associate systems feature voice, direct manipulation, text, WIMP, gesture, and/or visual input systems.

The Soar Training Expert for Virtual Environments (STEVE) is a virtual environment based associate system. STEVE is a teaching agent
for a virtual submarine. This agent has the ability to monitor and modify the state of the virtual environment. STEVE is spatially aware of the student and the virtual environment. The agent's tasks include construction, execution, and revision of lesson plans for teaching users to perform tasks in the virtual submarine [Rickel97] [Johnson98].

Steve's predecessor is Homer. Homer also performs simulated submarine operations. Homer is able to construct, execute, and revise plans that feature temporal reasoning. Homer can accomplish object avoidance in the context of submarine operations. Homer also features natural language processing to assist user interaction [Vere90].

Another category of intelligent user interface includes information retrieval or recommender systems. These systems serve to find appropriate information for a user's queries. One agent, ProfBuilder, inhabits a web site and assists external users with queries. The ProfBuilder system extracts a users web browsing patterns to recommend additional websites. ProfBuilder develops a probabilistic model of the content of a user's visited web pages. ProfBuilder then searches for relevant pages with similar content properties [Wasfi99]. Wasfi indicates that ProfBuilder combines both content-based filtering and collaborative filtering [Wasfi99] [Lieberman99].

Letizia, developed at the MIT Media Laboratory, performs similar functions as ProfBuilder. Letizia operates while the user is idle to search for related web pages. Another tool, SiteHelper, is also similar to ProfBuilder but it requires the user to explicitly specify and rate search keywords [Ngu97]. Two other web tools are Ringo and Fab. Ringo assists users searching for music. Fab recommends websites sites that might interest the user. Fab performs content analysis on previously visited sites and then requests the user to rank its suggestions [Wasfi99]. A non-web related agent, Ringo, helps users find appealing music. Ringo collects user ratings of music samples. The ratings are then compared among existing user profiles to recommend other selections the user might enjoy [Shardanand95]. The Lumiere interaction assistant developed by Microsoft Research, system studies user actions over time to determine the user's goals and needs. Lumiere combines Bayesian modeling techniques with probabilistic analysis of user
queries to deduce the informational needs of users [Birnbaum97] [Horvitz98].

The next category of interface we discuss is data presentation assistants. Two examples, SAGE and AIMI automate data presentation tasks. AIMI adapts the presentation of text, maps, or tables as appropriate for the user [Burger93][Stephanidis97]. SAGE characterizes and then displays data sets of diverse origins [Sage00][Birnbaum97].

Existing Intelligent User Interface agents vary in size and scope. This section has highlighted associate systems, information retrieval assistants, and data presentation agents. Many of the user modeling and prediction techniques used by these agents are applicable to the development of palmtop interaction devices. These systems provide a good cross section of existing intelligent interface agents.

This concludes the discussion of intelligent user interfaces. This section has presented motivations for intelligent interfaces. We have established the necessary criteria for and role of intelligent agents. A number of effectiveness issues have been summarized and a number of example agents have been listed.

Conclusion

There are a number of existing concepts and technologies that can enhance the design of a palmtop interaction device. The three main areas of related research include interaction, usability, and intelligent user interfaces.

This chapter has included a discussion of interaction systems. Both immersive and non-immersive interaction methods were discussed. That discussion reviewed existing interaction methods and compared the method's strengths and weaknesses. A palmtop interaction device used inside of the virtual environment features attributes of both immersive and non-immersive interactions.

The field of usability provides good methods for developing and evaluating systems. Usability can be applied to user interface development as well. The large number of potential uses for the palmtop interaction device will require careful analysis methods to determine effectiveness.
Intelligent user interfaces are one potential method of increasing a system’s usability. The IUI discussion presented the criteria, capabilities, and caveats of intelligent agent development. The palmtop interaction device potentially has available to it the user input patterns in the virtual world and the input provided through the palmtop interface. Intelligent user interface methods can potentially be applied to a number of palmtop user interaction tasks.

In the next chapter, we will discuss the palmtop interaction device for the virtual environment. The chapter will address how this device incorporates existing interaction methods. How usability methods are used to develop and evaluate the device. We will not use intelligent user interface concepts in the system developed for this thesis project, however we hope the reader will see the potential for automating the user’s tasks in the virtual environment.
CHAPTER THREE

PALMTOP INTERACTION DEVICE FOR IMMERSIVE VIRTUAL ENVIRONMENTS

This chapter presents the Palmtop Interaction Device for Immersive Environments. The first section discusses the concept for the palmtop interaction device. The second section details the system design. The final section for this chapter will discuss the iterative user test procedures used during the development of the interface.

Concept

The discussion of the palmtop interaction device system concept is divided into four sections: Motivations, Interaction Tasks, Applications, and Related Research.

Motivations

The development of the palmtop interaction system was motivated by my personal experiences in developing VR applications and hardware. In the last four years as a research assistant for ISU’s Virtual Reality Applications Center I have observed several needs in virtual reality systems. This section will discuss my observations as a VR application and hardware developer. The observations will then be categorized and discussed with relevant interaction information presented in previous chapters.

Experiences in Application Development

I have developed and contributed to several inter-disciplinary virtual environment projects. These applications have included architectural design, oil exploration, and application integration to name a few. These applications are briefly described below followed by a discussion of their interaction systems’ effectiveness.

VADeT. The Virtual Architecture Design Tool VADeT [Chan99][Hill99][Hill99b] was designed to test the feasibility of performing architecture design in an immersive virtual environment.
VADeT users are able to create and modify architectural design primitives. Users have the ability transform (translate, rotate, and scale), color, texture, or delete design primitives. Users can explore their designs by navigating or scaling the design space. Users can also review the history of the design project to evaluate decisions that were made. Interaction in VADeT is performed using a three-button wand device. Actions are performed via ray-selected two-dimensional menus, direct object manipulation, proximity and ray casting object selection, and a 2D Virtual User Interface Color selection panel [Heath98]. User feedback consists of a floating head's up display that indicates the status of the current operation.

Shell Oil. The Shell Immersive Oil Exploration Environment provided subsurface modeling, well planning, and seismic data interpretation functions in the C2. The seismic data interpretation application allowed users to load seismic survey information and to modify the appearance of the seismic data. This application also uses a three-button interaction wand and relies primarily on 2D Virtual User Interface [Heath98] to perform operations. One of the three wand buttons performed all VUI menu operation selections. The other two buttons would serve for navigation or editing purposes according to the task at hand.

Application Switcher. The application switcher was my first project at VRAC. This switcher, called the Hallway, featured a rectangular room with six large texture mapped portals. Each portal represented a separate virtual environment. Users interacted with the wand device to navigate toward and activate a portal. Upon activation, a transformation sequence placed the user in the newly selected environment.

Various Projects. Other VR application development projects have included a novel interface system called the Chordboard. This system provided a simulated six-degree of freedom pen and tablet interaction system. This system used dual tracked input devices. One tracker sensor was attached to the pen and another was attached to an 18 in. by 12 in.
sheet of transparent Plexiglas. The Plexiglas provided a flat surface for drawing on or selecting operations in the virtual world. A four button input device was attached to the bottom surface of the board. A textured flat polygon was drawn in the virtual environment, coplanar to the chord board, to generate the simulated 2D display.

**Interaction System Effectiveness**

User tests of VADeT application indicated that users can easily understand the operation of the menus, however a good amount of motor control is required to 'aim' for an operation while simultaneously pressing a wand button. Hierarchical 2D menus present an extra challenge to the user because it was not always clear which menu operations would lead to additional menus and which menu items caused actions to be performed. Another difficulty arose when users performed ray selection of objects. In some cases inadvertent selections would occur. Unfortunately the users would perform modifications on the incorrectly selected shape before realizing which object was actually selected. The application switcher used only proximity selection to activate portals. While there was a significantly lower chance of inadvertent selection, users were required to navigate toward any object that was beyond their 'reach' in the C2.

Input device functional overload was very high in VADeT. Button operations were chosen to be as consistent as possible and to assign similar mode sensitive functions to the same buttons. For example: one button performed all object editing operations, one button performed all object selection, and the last button performed all menu operations. This assignment model worked well until more complex operations like shape grouping or design playback capabilities were selected. Users reported difficulty remembering the mapping between wand buttons and virtual environment operations.

One remedy for the VADeT interaction shortcomings could have been to improve the environmental queues or to modify the operation of the virtual controls. For example, introducing a virtual ‘scrolling’ cursor that would allow the user to traverse the menu operations could reduce the menu selection difficulty. The scrolling cursor however would not convey the menu’s hierarchical structure to the user. Displaying wand
button mappings in the virtual environment could reduce user confusion regarding wand button overloading.

An issue to note is that both the scrolling menus and the wand button mapping display would introduce new visual elements into the virtual world. The floating heads-up-display (HUD) text was implemented with the similar goal of presenting information to the user. These solutions may correctly inform the user about the state of the virtual environment; however, displaying that information occludes some aspect of the virtual experience. With VADeT, the heads up information display was tracked to the users head and the information lie in the periphery of the user's view. User testing revealed that this information was difficult for some users to see. The users would often rotate their head to look toward the HUD, and the HUD would then rotate further away from the user to remain in their peripheral view. In the end most users seemed to ignore this information display. From this observation, we can safely say that floating information displays must be placed correctly to be readable by the user and not distract from the virtual experience.

The Shell Interactive Oil Exploration Project presented a different set of difficulties regarding 2D virtual user interface windows. The Virtual User Interface (VUI) project allows for users to place windows at arbitrary locations in the virtual space. Placing and Interacting with a single VUI window was a simple process. Unfortunately, the size of windows, typically over 2 feet in width and height, impacted the visibility of the oil field. The overall VUI window size could not be reduced without sacrificing panel readability. Many of the Shell project menu options were hierarchical. VUI does not support hierarchical menus within the same window; instead, new windows are opened near the users location. This often led to a number of open menus simultaneously occupying the same spatial coordinates, intersecting, or in some cases overlapping each other. As a result, it became difficult to select or interact with the menus without first moving them. Moving the windows to different locations then obscured more of the oil field.

The Shell seismic interpretation application allowed users to modify color map values through a 2D interface. From an interaction
perspective, this task is similar to drawing a curve on a 2D window at an arbitrary orientation in the virtual space. This task, while easy to understand, was somewhat difficult to perform. The lack of a physical 2D surface allowed the interaction device to move in and out of the window’s depth. Also, users needed to stand within arms reach of the window to interact with the data. At this distance it became difficult to focus on the window or on the color map curve being drawn on the screen.

The Shell interaction system also overloaded the wand button functions. The amount of overload is not as severe as with VADeT, however an incorrect button selection often resulted in opening a VUI window or activating the velocity based navigation system.

Both the Shell and VADeT project could have been improved by adding the ability to specify exact numerical values. This could have improved object transformations in VADeT and oil field editing in the Shell project. The Shell project also stored a vast amount of alphanumeric data about objects in the virtual environment. Unfortunately, we did not have an easy method for accessing and modifying these variables and were not able to implement appropriate interactions for them.

The Chordboard interaction system was very preliminary in nature, but it was clear that the flat 2D plane helped with selecting items on the 2D simulated display. This project suffered from occlusion whenever the users body was positioned between the Plexiglas and the projection surface. Another issue also indicated the importance of accurately registering the graphics to the Plexiglas display. Tracker latency, calibration errors, or any graphics frame pipelining would seriously degrade the performance of the simulated display.

Experiences in VR Hardware Development

My experiences in developing VR hardware have included two main efforts. My Computer Engineering undergraduate senior design project was the development of an integrated wand and LCD display device for immersive projection technology systems. The second effort has been as team leader for the C6 input device group. These experiences are discussed in order below.
**LCD-Wand.** The LCD-Wand was developed to examine the possibilities of combining digital and analog inputs with a hardware graphical display system inside of immersive projection technology environments. Essentially this device pairs a three-button mouse/joystick device with an LCD panel and six degree-of-freedom tracking. The main goal was to provide the user with an additional input/output channel while interacting.

The LCD-Wand project was similar to the palmtop interaction device in that both projects wished to present the user with additional help or interaction information through the display. This project faced a number of issues due to the state of the technology at the time. Palmtop computing was not a viable option at the time, so our group made customized hardware including a 64x128 pixel black and white backlit display. The information we gained from this project was enough to encourage future study into this type of interaction device.

**C6 Input Device Group.** This team was responsible for selecting and developing wireless input devices for the C6. This team surveyed a number of potential input devices. This group was fortunate to begin its search around the time that wireless RF computer peripherals came into the limelight. The final selection was a PS2/serial compatible device with three mouse buttons and an integrated thumb-joystick pad.

This section has only indicated a small amount of the knowledge gained from the research projects. The next section will summarize additional needs that have been identified in other projects, written in literature, or expressed by fellow researchers.

**Interaction Needs**

We now present a summary of interaction needs that have been observed in research projects, literature, or in conversation with other VR developers. One interaction need concerns alphanumeric text input methods. The author feels that it is necessary to provide a more efficient means for specifying annotations and decimal values for virtual objects. At present, alphanumeric data can be specified through traditional keyboards, virtual interface keyboards, and gesture or
speech recognition systems. As stated earlier, keyboards are not well suited for interaction in the virtual environment. Virtual interface keyboards can obscure the virtual world and do not provide physical interaction constraints. Virtual user interface window sizes require users to move over a large area, which can increase interaction time and lead to user fatigue. Gesture and speech interaction tools can misinterpret actions or user speech. Gestures require time to learn and performing them can lead to user fatigue.

The Palmtop computer allows VE designers to integrate standard UI components into the virtual environment. This integration helps to reduce the learning curve for the device. The interface elements are familiar to the user so little cognitive effort is required to comprehend the 2D controls or their operation. Additionally, the appearance and operation of the controls are uniform across applications. This reduces the training time required for the user.

There is a need for additional media capabilities in virtual environments. Vocal annotations and speech recognition systems require audio input mechanisms. Sound output requires an audio channel. Some images require high-resolution detailed imaging in the virtual world, which may not be available due to VR visual acuity limitations. Video conferencing applications and avatar representations would only benefit from increased visual quality. Further it would be beneficial to provide all of these capabilities to the user through a single device. Palmtop computers are portable encapsulations of these capabilities.

Palmtop computers typically feature integrated microphones and speakers that can be used in the virtual environment. The palmtop’s recording capabilities can facilitate verbal annotations. The output channel can facilitate collaborative efforts by providing an additional point-to-point audio channel. In multi-user CAVE settings, the palmtop audio channel allows messages to be sent directly to the intended recipient. Palmtop computers also feature high-resolution displays in excess of 320x240 pixels at 65,536 colors. This enables the palmtop for displaying images or video. A palmtop computer can be used to implement point-to-point video conferencing in the virtual environment or it could be used to examine details in graphic imagery.
Palmtop computers are small and easy to manipulate. A palmtop can be connected to the Internet with a single cable. Wireless radio frequency networking options are increasingly available with transmitters providing ranges in the hundreds of feet with megabit per second data rates. The palmtop is designed for low power consumption and features a self-contained power source. These wireless capabilities make the palmtop idea for integration into the virtual environment.

Palmtop Interaction Tasks

We have discussed the need for alphanumeric text input, standardized UI components, media capabilities, and easy to manipulate devices. Now we shall identify which interaction tasks this palmtop computer can be used for. Each task is listed below, followed with a brief description of this type of interaction.

Alphanumeric Information

The palmtop computer’s touch sensitive display enables character recognition software for text input. Palmtops often offer simulated keyboards that can be activated from the system tray located at the bottom of the display. Alphanumeric information can be used to annotate virtual objects, set object attributes (position, orientation, color), issue commands, or send messages to other VE inhabitants.

Indirect Object Manipulation

Virtual Objects might be interacted with through direct or indirect means. Direct forms of interaction include wand or glove selection and placement. Indirect methods would include using 2D knobs, or sliders to modify object attributes. A set of 2D sliders could be used to set position or orientation for virtual objects. A simulated 2D track pad area could be used to ‘slide’ objects along virtual axes.

Help Information

An application could present a large amount of text help information to the user via the palmtop display. The palmtop display resolution offers improved readability over virtual text. Further, the
palmtop text does not interfere with the imagery of the virtual world. Potentially, the entire Internet is accessible through the palmtop device. This allows the VE user to have access to a wealth of information inside the virtual environment and it enables application designers to use the multimedia capabilities of the Internet.

**Way Finding**

The palmtop device could display map information of the virtual environment for the user. The device becomes an in-hand 'you are here' guide. The device could display a low-res model of the virtual world. It could indicate locations of markers, annotations, or other users in the virtual world. This device might also be used to indicate beginning and endpoints for a path planning system.

**Navigation**

The palmtop device could be used to select a direction of motion or velocity for navigation tasks. Watsen presents a system for performing such navigation in a palmtop device. The interaction device is used to select heading and elevation from a two-dimensional plane. The input device also allows for basic velocity settings (forward, reverse, stop) [Watsen99]. Any number of navigation constraints or scale factors could be set through the palmtop device.

**Operation Selection**

Palmtop buttons, menus, and dials could be used to perform operations in the virtual environment. LED-style buttons could indicate VE state to the user. Palmtop UI menus would behave similarly to controls found in contemporary WIMP interaction systems.

**Appearance**

Object color, texture, or material properties could be selected, enabled, or disabled through a palmtop device. Lighting parameters could also be selected through the interface. The palmtop’s color display also provides a suitable method for viewing the appearance of a user’s material, color, and lighting selections.
Object Selection/Identifications

The palmtop could serve to identify and select virtual objects. Objects could be selected from a list of all existing virtual objects. Shapes can also be chosen via proximity or ray-casting methods. Once chosen, the palmtop could display relevant attributes for the device. In this capacity the palmtop becomes a sort of Virtual Tricorder [Wloka95]. With this we conclude our discussion of possible palmtop interaction types.

Applications of the Palmtop Interaction Device

The Palmtop interaction device has a number of potential applications. The interaction functions listed above are performed in VR-CAD, Art, Design, Data Visualization, Virtual Prototyping, and Manufacturing virtual environments. Here we further discuss potential palmtop interaction methods in these application areas.

VR-CAD

Virtual Reality Computer Aided Design systems are used to perform all of the interactions listed above. These systems feature extensive model exploration and geometry creation. A further complication with these systems is the abundance of operations that need to be performed. Care must be taken to organize all operations in an intelligent manner [Chan99][Hill99][Hill99b].

Art/Design

Art and Design visualization systems utilize material property selection methods. A system could potentially offer gigabytes worth of texture information. This information would overwhelm the memory requirements of a palmtop if displayed simultaneously, however the Internet provides an efficient method for browsing subsets of this texture information.

Data Visualization

The task of visualizing multidimensional datasets is enhanced by graphical capabilities the virtual environment. This task could
integrate the palmtop computer to identify and view attributes for the data points. The Palmtop can be used to quickly specify which sub-set of the multidimensional data we wish to examine at a given time or to select axes for a grand tour animation of the data.

**Virtual Prototyping**

Simulations of prototype designs can benefit from palmtop interaction. The palmtop might also be used to select alternate parts from a part database. The palmtop can be used to enable or disable the view of stress/strain information in the virtual part set. It can serve to annotate observations in the VE. It could function as a web utility for interfacing to company intranet sites to locate a part’s history.

**Manufacturing**

Factory simulations can incorporate the Palmtop device to observe or modify simulation parameters. The Palmtop could show the history of a manufactured part. Selecting manufacturing history events from the palmtop device could initiate an automatic review the part’s journey back to that point in time.

There are several more potential applications of a palmtop interaction device. In next section we will examine previous research efforts related to the use of palmtop computer interaction.

**Related Research**

The palmtop interaction device extends a number of previous interaction research efforts. This section will discuss a Palm Pilot interaction device for immersive projection technology systems, a spatially aware palmtop display for cubic spreadsheet interaction, a haptic palmtop interaction display, and a virtual Tricorder metaphor for interaction and menu selection.

Kent Watsen integrated a PalmPilot™ computer with a CAVE-like environment. Watsen’s design includes a multi-tabbed applet window with camera, environment, and geometry controls. The operation of these controls is discussed below [Watsen99].
The camera applet facilitates navigation. The X and Y pointer position indicates heading and pitch respectively. On screen buttons allow the user to toggle the navigation mode between forward, neutral, and backward motion. The environment panel is used for color selection. The geometry applet features six horizontal sliders for adjusting XYZ position and HPR orientation of an object. This applet also features a tab for selecting the current object [Watsen99].

The palmtop interface was developed as an extension to the Bamboo networked virtual environment software library to simplify integration into existing projects. Watsen indicates that a future step for the project would be to integrate tracking information with the palmtop. This tracking information would allow direct manipulation of virtual objects and would open the door for spatially aware interaction metaphors [Watsen99].

Another related project is the Chameleon. Chameleon is a spatially aware palmtop display. This research was conducted in the early 1990's before the development of current palmtop technology. Instead, graphics are generated by a workstation and displayed on a miniature hand held television screen. Spatial awareness is implemented with a six degree-of-freedom motion sensor that is attached to the display [Fitzmaurice93][Fitzmaurice94].

The Chameleon is used to navigate through an egocentric virtual cubic spreadsheet. The user traverses the spreadsheet by manipulating the display. The author presents a number of navigation methods. In one method, tilt-and-go, users angle the display in the desired traversal direction. The degree of tilt indicates the user's velocity. In another method, individual buttons on the device indicate the desired direction of motion [Fitzmaurice93][Fitzmaurice94].

Noma et. al., present a palmtop display with haptic feedback for performing virtual space exploration. The display is mounted on a force-feedback mechanical arm. The user manipulates the display to view virtual objects. The authors present a number of interaction methods and discuss the results of a user test experiment [Noma96].

Wloka and Greenfield presented the Virtual Tricorder (VT) as a uniform VR interface in 1995. While this effort does not involve a palmtop display or a palmtop computer, it functions similarly to a real
palmtop interaction device. The VT consists of a sonically tracked Logitech flying-mouse. A virtual two-dimensional menu is drawn just above the mouse position. A Boom or a HMD display displays both the mouse and the menu [Wloka95].

Two uses of the VT are to perform menu selection and thruster-like navigation. The VT selects objects by projecting a selection cone into the virtual space. The VT can also operate as a magnifying glass for zooming in on virtual artifacts [Wloka95].

The authors note that visual quality is a concern with this system. HMD resolution at the time was limited. In their experience, users would pull the Tricorder close to their face to increase readability [Wloka95]. The Virtual Tricorder is familiar to Star Trek® fans; however the design is not compatible with immersive projection technology systems. This system requires a HMD or enclosed display to prevent stereo violation or occlusion that would occur in projection based systems. Additional requirements include low tracking latency and high tracking accuracy to preserve registration between the virtual menu and the mouse position. With this application, we end our discussion of related palmtop interaction efforts.

This section has presented motivations for developing a palmtop interaction device. We have discussed my personal experiences in VR technology. We have seen a number of interaction tasks and applications that could use this device. We have also surveyed research projects that are related in interaction or design to this one.

**Design Details**

In this section we present the software and hardware design details associated with the palmtop interaction device. We begin with a discussion of the hardware platform followed by a description of our palmtop software system; the Java based Interface to the Virtual Environment (JAIVE).
Hardware Requirements

The hardware overview presents the requirements for the systems include both the VR system and the palmtop device. The hardware overview will also detail recommended palmtop characteristics including the processor, memory, display, controls, audio, and communications subsystems.

**VR System**

The palmtop interaction system presented in this thesis is designed for use in Immersive Projection, Power-wall, or Desktop VR systems. The placement of projection surfaces in projection-based and desktop systems allows the use of physical objects in the virtual environment. This is not the case in enclosed head mounted display systems where the projection surfaces are inches away from the user. An obvious exception would be augmented reality systems with transparent or semi-transparent projection surfaces.

**Interface Device**

This system also requires a palmtop or tablet sized computing device. There are a number of devices on the market with impressive characteristics [Palm00][Jornada00][Cassiopeia00][Qbe00]. The specific requirements for this system include portability and wireless operability.

The desired portability characteristics include low weight and good ergonomics. Essentially we want a device that will not hinder the user's mobility or cause discomfort over sustained periods of use. To meet the wireless operability characteristic, the system should have self-contained power supply and wireless networking. Most palmtop computing devices include self-contained power. Wireless networking options are also available. The performance for these cards ranges from wireless-modems to wireless-Ethernet cards. Most of these cards are designed for the PCMCIA card slots found on a number of laptop, tablet, and palmtop computers.
Recommended Hardware Configuration

There are a number of palmtop computing solutions available. The solutions span a wide range of processing, memory, display, and other capabilities. This section identifies the performance characteristics this system has been designed for.

Processor. The JAIVE system is sufficiently responsive on a Pentium II class machine. Processing speed on a Pentium II should be above 200 MHz. The user interface should update at least four times per second to support user interaction with the virtual world.

Memory. This system should contain sufficient memory to support the platform specific Java Virtual Machine. Note that the memory requirements are also related to the amount and type of visual content the JAIVE system must display.

Display. The hardware device should support a minimum 640X480 pixel display with at least 256 colors. Brightness is also a desired characteristic for the device. The display should be easily readable in a dark or unlit room. The display should also either be non-polarized or polarized such that it is readable by a user wearing a set of active shutter glasses.

User Controls. The device should have a touch sensitive screen that will allow the user to use a finger or a stylus to select and manipulate the user interface. The device should have a mouse button to indicate mouse click or drag operations. These controls should be placed to allow use by right or left handed individuals. The system should provide a keyboard or simulate a keyboard on the display for alphanumeric data entry.

Audio. The device should feature speakers and a microphone. A microphone could be used to record annotations, speak to other users in remote locations, or to facilitate vocal commands. The speakers would allow playback of annotations, audio feedback, or other audio cues related to the interface or the virtual environment.
Communications. The system should provide wireless network connectivity to the Internet. The latency and bandwidth should be sufficient to allow interactive UI operations and high bandwidth multimedia capabilities. Network characteristics should meet or exceed the user’s expectations of a comparable 2D workstation.

Software Design - Java based Interface to the Virtual Environment

The Java based Interface to the Virtual Environment (JAIVE) system architecture is detailed below [Hill00]. We will present a brief introduction to the JAIVE system including the system design principles and the software platform. The remainder of this section details the software architecture components. The architecture discussion is subdivided into JAVA to C++ communications, JAIVE control panels, JAIVE support classes, and JAIVE callbacks.

Brief Overview

JAIVE is a collection of user interface control panels implemented in Java. These panels are presented in a multi-tabbed user interface window, which executes on the palmtop computer. The Java interface panels are configured through a C++ server. The system has a text feedback window that displays the most recent user operation. The interface also has a top-level pull-down menu for exiting the application. Figure 3.1 depicts two JAIVE users in the C2.

Figure 3.1 JAIVE Users in the C2
The JAIVE control panels include:

- Three degree-of-freedom interaction
- Operations/selection menu
- Text annotation
- Information display
- Map display
- Color selection
- Parameter adjustment
- Picture display

The state of the Java panels and any user operations are 'reflected' by a set of C++ classes. Any operations on the Java panels are routed through the communications subsystem to the corresponding C++ panel-mirroring object. Similarly, any operation on the C++ mirroring object affects the corresponding the Java panel. The C++ mirror component notifies the user application of a change through a user specified callback function. Alternatively, the C++ application may poll the interface mirror to determine a component's state.

JAIVE networking is implemented with TCP/IP sockets. JAIVE defines a message format that supports JAIVE system operations and parameter modifications. The JAIVE messages, message queues, and networking comprise the applications communications subsystem.

The JAIVE panels, JAIVE communications, support classes, and callback capabilities will be discussed in the later half of this section.

**Design Principles**

JAIVE's interface design draws upon the usability design principles discussed in chapter two. The chief usability issues addressed in this system's design are listed below.

- Maximize Visibility - the multi-tabbed window provides the user with a constant view of all available high-level panels. The feedback window also updates to show the most recently performed operation.
- Accommodate User Error - this system does not directly provide undo methods, however the interface allows the C++ application to 'reset' the state of all interface elements and variables if the user indicates that an error has occurred.

- Use Familiar UI Tools - All Java user interface elements are standardized similar components. These components are ideal for quickly developing WIMP UI applications. The JAIVE system utilizes check boxes, push buttons, a multi-tabbed interface, sliders, pull-down menus, text entry fields, and a multi-modal color-selection panel.

- Correct Mappings - The JAIVE system clearly and explicitly labels all UI controls. Push buttons are labeled by exact function. Sliders values are reflected in text fields. Some panels feature alternate arrangements of UI elements to allow users to select the control layout they are most comfortable with.

- Exploit User Knowledge - The familiar UI controls reduce the amount of time users require to learn basic application interactions. The user's prior experience with two-dimensional UI applications transfers directly into JAIVE interactions.

- Use Constraints - All sliders are labeled with maximum and minimum value points. The slider extents are fully configurable and Java ensures that slider constraints are observed consistently. All open interaction areas also indicate when users have exceeded the boundaries of the interaction space.

- Simplify Access to Often Used Commands - JAIVE provides the user with quick visible access to all interface elements. In general, panels containing multi-tabbed sub-panels retain previous state even when they are not exposed. In other words, multi-tabbed sub-panels continue to display the same sub-panel even when they are not actively manipulated.

- Easy to Integrate Software - An important goal of this project is to provide an easy to use software-programming interface. We felt it important that the developer be able to control the
amount of integration between the JAIVE system and the VR application. The developer should be free to enable or disable this interface, or to use other interfaces in its stead. This is accomplished by providing the developer with both callback and polling methods to determine the UI state. Further, aside from the JAIVE interface mirror elements, and callback operations the user is not required to utilize any JAIVE support classes to store his/her data.

Software Platform

The JAIVE system is developed using the Java programming language. Java is a good choice for developing this project for a number of reasons. One reason for this selection is the multi-platform nature of the language. Java Virtual Machines are available for several operating systems including: Windows, MacOS, UNIX, IRIX, and to a limited extent PalmOS and Windows CE (now Pocket PC).

Java also provides easy to use UI elements that can be created and configured at runtime. Java 2 Swing components maintain similar appearance across platforms. Java also provides nicely abstracted TCP/IP communications, rich multimedia capabilities, integrated WWW resources, exception handling, and integrated debugging capabilities.

This concludes the general discussion of the system, the basic design principles, and the motivations for selecting the software platform. In the next section we shall further examine the specifics of the JAIVE software.

System Overview

We now discuss the specifics of the JAIVE system. This section introduces the Java/C++ interface components - JiMirror and JiInterface. The next topic in the section will be the communications subsystem, followed by a listing of the JAIVE control panels, support classes, and callback facility.

Java/C++ Interface. JAIVE provides two main classes that are responsible for interfacing the C++ application with the Java user
interface controls. The Java class JiInterface and its companion C++ class JiMirror are the center of the JAIVE system.

JiMirror 'reflects' the state of JiInterface. The JiMirror class provides the developer with options to create and configure JiInterface panels. JiMirror reflects the state of the JiInterface panels. Figure 3.2 indicates the relationship between the user application, JiMirror, JiInterface, JiComponents and the interaction computer.

![Figure 3.2 JiMirror and JiInterface Block Diagram](image)

JiInterface is the central access point for all interface panels. JiInterface is responsible for creating any interface panels that are added to the JiMirror. JiInterface also communicates any panel status changes or user operations to JiMirror.

Communications. All communications between JiInterface and JiMirror occur through the JiCom subsystem. JiCom provides connection oriented and datagram communications between C++ and Java applications. JiCom uses a number of related subclasses including JiServer, JiClient, JiMessage, and JiMessageQueue. These classes are listed and their functions described below.

- **JiServer** [Vertanen99]
  - Create a TCP/IP socket server in either Java or C++
  - Send and receive connection oriented or datagram data as character, integer, float, string, or byte over connection
- Provide transmission receipt acknowledgements for connection oriented communications
- Translate data byte ordering between C++ and Java data types
  - JiClient* [Vertanen99]
    - Connect to a JiServer from Java or C++
    - Send and receive data
    - Provide acknowledgements for data communication
    - Translate data byte ordering
  - JiMessage
    - Generic JAIVE message format
    - Support destination and content fields
    - Parse messages and generate message tokens for use by receiving component.
  - JiMessageQueue
    - Short term storage for JiMessages between receipt from JiCom and dispatch to interface panel classes.
    - Provides First-In First-Out access to messages

* JiServer and JiClient are strongly based on Keith Vertanen’s C++/Java socket server class. This source code has been made freely available by the author via the Internet [Vertanen99].

Upon JAIVE system startup the C++ JiMirror creates a JiCom object that then creates a JiServer object. JiInterface instantiates a JiCom object that then creates JiClient object and connects itself to the JiMirror server. Each JiCom class creates an incoming and outgoing JiMessageQueue. At present, all outgoing data transfers are non-blocking and occur immediately upon invocation. Incoming data reads are also non-blocking, however the JiMirror processes messages only when invoked explicitly by the user’s application.

In the message-processing step, the JiMirror or JiInterface extracts a message from the JiMessageQueue. The message is broken up into tokens, which are then routed to the appropriate destination JiBaseComponent.
JiCom is also used to provide synchronization between the JiInterface and JiMirror. This is accomplished with the JiServer/JiClient send/receive acknowledgement operations. The C++ JiCom class sends a single blocking connection oriented byte to the Java JiCom class. The C++ application remains frozen until a reply is received from the Java interface. The first-in first-out queue processing ensures that the Java interface has processed all other JiMessages up to that point.

Control Panels. JAIVE provides several pre-made interface panels. These panels provide basic interaction functionality and allow developers to customize data associated with the panel. Every panel has both a C++ mirroring component and a JAVA interface component. All JAIVE interface panels are derived from a base component JiBaseComponent. JiBaseComponent is responsible for handling inter-component communications through the JiCom subsystem. JiBaseComponent also handles panel naming and basic UI panel appearance including background color and panel status. The interface panels and their descriptions are discussed below.

Ji3dof - This panel provides generalized three degree of freedom interaction (Figures 3.3 and 3.4). Positional data can be specified through a trio of sliders, a differential touch window, or through alphanumeric entry. Developers can customize all panel labels, slider maximum and minimums, and slider text field labels. The Ji3dof panel contains four different layout options - horizontal, vertical, arranged, or touch panel. The two arrangements of these sliders are shown below.

JiOps - This panel implements user action input selection through button menus (Figure 3.5). Operations can be added to a single panel or groups of operations can be formed in which case each group occupies a tab within the panel. Operation buttons can be labeled with text, by color, and/or with image icon.
Figure 3.3 Ji3dof Horizontal Slider Arrangement

Figure 3.4 Ji3dof Touch Panel Interface
JiNote - JiNote is used to create and modify text annotations. JiNote features an editable text window, an annotation selection menu, an annotation title field, and buttons for creating, committing, or deleting annotations. The corresponding C++ component provides methods for adding or modifying JiNote information (Figure 3.6).
JiInfo - JiInfo provides help information to the application user. This panel operates very similarly to JiNote. JiInfo features a topic selection window, a title window, and a non-editable text (Figure 3.7) or HTML viewer (Figure 3.8). Java provides inherent URL loading capabilities provided with basic HTML tag support. This panel can be used to display help content directly from the Internet.

Figure 3.7 JiInfo Text Viewer

Figure 3.8 JiInfo HTML Viewer
JiMap - JiMap is capable of loading image files and monitors the JAIVE cursor's position in terms of the picture coordinates (Figure 3.9). This panel could potentially support overhead map view navigation of virtual spaces. Another potential use is to display the positions of objects or individuals in the virtual world. This panel features the image display area, cursor appearance selection menu, and mouse X, Y position text fields.

Figure 3.9 JiMap

JiColor - JiColor presents the user with the default JAVA color selector (Figure 3.10). This panel is comparable to PC or MacOS color selection tools. Colors may be selected from a color swatch palette, through RGB (red, green, and blue) sliders, or through HSV (hue, saturation, and value) sliders.

JiParam - JiParam allows users to select and modify numerical or toggle binary parameters associated with the application (Figure 3.11). This panel features a parameter selection menu, an adjustable slider, a numeric text input window, a binary toggle, and a parameter commit button. The JiParam C++ interface allows the developer to add new parameters, indicate the type of data associated with the parameter (binary or numerical), and to set the maxim and minimum slider values.
JiPicture - JiPicture is a simple picture viewer. This panel can load files directly from the local file system or over the Internet. This panel has potential for displaying customized virtual environment data on the palmtop device.

We now conclude the overview of the JAIVE control panels. These panels represent the full JAIVE interaction set. The set listed above contains all of the panels we will use for this evaluation exercise.

Support Classes. The JAIVE system implements a few non-panel specific support classes. Those classes are JiVec2, JiVec3, JiString, and JiStringList. JiVec2 and JiVec3 are simple vector classes. JiString is a text string class. JiStringList implements support for string collections.
Callback Interface. The JAIVE system features a simple callback interface that alerts the user application when the JAIVE state has been changed. The user application’s callback function receives an event object that indicates which operation has been performed.

The JAIVE callbacks are implemented through two classes: JiCallback and JiEvent. The JiCallback object accepts a pointer to a user function with a JiEvent object as a parameter. The JiEvent object contains two information fields: component and event. The component field indicates which component has issued the callback and event indicates which operation was performed by the issuing component.

Each JiBaseComponent creates a JiCallback object. The JAIVE control panel will invoke this callback object whenever new status updates have been sent to the component. The user application can hand a single callback function to all JAIVE control panels or have one callback per control panel. The JiEvent fields serve to differentiate the calling control panels and determine the appropriate action for the user application.

JAIVE Section Summary

The JAIVE software system is the beginning of a cross-platform palmtop user interface library. This system has been designed to allow quick and easy integration into existing applications. This is accomplished by providing the software developer with a number of commonly needed predefined panels. These panels are all configurable.
to some extent from the C++ application and they allow developers to derive new panel's features as needed. The next task is to determine how well the JAIVE system operates with VR applications. The user testing method is described in the following section.

User Testing

The palmtop interaction device's effectiveness is evaluated using iterative usability methods. This section details the iterative usability method, the test methodology, and the data collection and analysis procedure.

Iterative Usability Testing

The goal of the usability testing is to judge users' performance with the JAIVE system. User's speed and accuracy in completing typical VR interaction tasks will be examined. Also, survey methods will be used to examine users' attitudes toward the different aspects of the interaction systems.

Methodology

The testing cycle was conducted in two phases. All phases of the testing were done using the Virtual Reality Application's Center's C2 or C6 Immersive Projection Technology systems. Each phase featured a test application in which the participant is required to perform some task(s) using the JAIVE system executing on a tablet computer. The experimenter recorded the participants' performance time and accuracy. The experimenter also noted any of the participants' observable qualitative behaviors. Participants were asked to complete survey forms about their experience after completing the task(s). An additional general survey was given to participants to determine background information relevant background information. The background survey and the first, second, and third testing phase procedures are described below.
Background Survey

The background survey (Appendix) was given to help us better understand our test subject population and to help us interpret individual responses. The participant group was asked a number of questions pertaining to their age, gender, vision, education, dominant hand, and English fluency. The group was requested to indicate their level of computer experience with PC, Macintosh, UNIX, Palmtop/Tablet, and Virtual Reality systems. Participants were requested to indicate which palmtop computer operating systems they were familiar with. Participants also indicated when their last virtual reality experience had been as well as which VR systems they had used in the past. Finally, participants were asked to judge their level of comfort with palmtop computers and with learning new applications.

First Phase

The first test phase required participants to perform tasks in two virtual modeling applications. Both modeling applications allow users to create and modify three-dimensional modeling primitives, but each application uses different interaction methods. One application, VADeT [Chan99][Hill99][Hill99b], solely used wand interaction methods. VADeT's interaction methods include ray-selected two-dimensional menus, wand based shape manipulation, and the Virtual User Interface's color window [Heath98]. The other application, written for the purposes of this study, featured a combination of JAIVE control panel operations and wand based shape manipulation methods. The first phase used the Aqcess technologies Qbe tablet computer as the JAIVE hardware device. The Qbe is pictured in Figure 3.3a and Figure 3.3b.

The first phase participants' tasks were to become acquainted with both modeling applications and then recreate a simple shape template with each application. Each shape template consisted of three rotated, scaled, colored, and/or textured shapes (Figure 3.15). The participants' first application was selected at random. Participants were allowed to ask as many questions as they required while performing the task. After creating both templates, users were asked to study the appearance and operation of two JAIVE control panels not specifically
Figure 3.13 Qbe User in the C2

Figure 3.14 Qbe Device [Access00]

Figure 3.15 JAIVE/Builder and VADeT Shape Templates
used in the test application. Finally, users were asked to complete survey forms about their experience.

The first phase survey (Appendix) form asked a number of questions about the interaction methods used to complete the tasks. The form asked for users to indicate a preference if any existed for the JAIVE or virtual interaction methods that were different between the two modeling applications. Participants answered questions about the readability and comprehensibility of the JAIVE control panels. Further, they answered questions about device ergonomics and the amount their view of the virtual world was obscured by the tablet computer and the virtual interface windows. Participants were invited to rank the panels according to how beneficial they felt the panels would be in virtual environment applications. Participants were also given the chance to elaborate on what would be an ideal device, suggest additional interaction panels, and write any additional observations they may have made about the system. The exact questions and user responses are discussed in chapter four.

Second Phase

The second test phase was designed to retest the deficient areas of the JAIVE system that were discovered during the first round of testing. The first round of user testing indicated that a different tablet/palmtop computer was required for users to be effective with the system. The second round of testing used the Intermec 6642 Pen*Key tablet computer pictured in Figure 3.16 [Intermec00]. The Intermec Pen*Key is a Windows 98 compatible PC with similar processing capabilities as the Qbe device. The chief benefits of the Intermec device over the Qbe for our specific application include smaller size, lighter weight, integrated numerical soft keys, and improved touch screen sensitivity. Another key advantage is that there is no conflicting display polarization with the shutter glasses as there was with the Qbe.

Another problem noted in the first test round was the limited granularity of the three degree-of-freedom panel sliders. These panels were modified to accommodate floating-point values out to two decimal positions of accuracy.
The second test round task required users to recreate the same design template used in the first test round using the JAIVE/builder application. This time however, users were given exact information about the position, orientation, and scale of the template shapes. This information was displayed in the JAIVE information help window as a simple HTML page. Users were instructed to only use the tablet computer during the exercise. They were also encouraged to use the Intermec soft keys to enter numerical data. Users did not perform any tasks with the VADeT application.

Second round participants were required to complete a survey (Appendix) form about their experience. This form asked participants about their understanding of the system, ease of entering data through the Intermec computer's numerical soft keys, and their impressions about the physical aspects of the Intermec computer in the virtual environment. The exact questions, user responses, and discussion of the results will be discussed in chapter four.

Data Collection and Analysis

All quantifiable survey questions were analyzed to determine average responses, deviation, and max/min ranges. The survey questions were also analyzed with the ANOVA and F-tests. The results of the data analysis are discussed in chapter four.
User Testing Section Summary

This section has described the experiments that we have used to evaluate the JAIVE system's usability. The first experiment compared user's performance in two applications with and without the JAIVE system interface. The problems discovered in the initial round of testing were fixed and then re-tested in the second test round. The second round of testing was designed to focus on a specific area of system improvement. In the next chapter we discuss the exact results of the usability test rounds.

Conclusion

This chapter has presented a number of aspects of our palmtop interaction study including the system concept, the system design details, and an overview of the usability study.

We have presented the thought process behind the development of the palmtop interaction methods including the author's motivations from personal experience and observed interaction needs in VR applications. Included in the device concept presentation we discussed a number of interaction tasks our system would try to include and which types of applications would exploit these features. Related research projects were also described as additional motivational factors in the development of this system. These related research projects ranged from pre-palmtop computer systems to software simulations of tablet style interaction methods.

The design detail discussion included a description of the hardware requirements of the JAIVE system ranging from the VR system characteristics to the palmtop computer hardware. The JAIVE software design was also presented. The software architecture decisions were discussed and a JAIVE system overview was presented.

The last section of this chapter presented an overview of our user testing procedure. The overview described the general plan for usability evaluation and presented the two rounds of user testing exercises. The overview also explains the target participant group of
experienced VR users and indicates how the user test data is used to drive subsequent rounds of user testing.

Chapter four will discuss the quantitative and qualitative data gathered during the usability testing. This data will be used to test and discuss our hypothesis about the effectiveness of palmtop interaction in immersive VR applications.
CHAPTER FOUR
RESULTS AND DISCUSSION

In this chapter we discuss the exact findings of the JAIVE user testing procedures outlined in Section 3.3. The chapter will begin by describing our participant group. Next, each test round will be discussed. For each round of testing we shall describe the data sample, present any experimenter observations, describe the participants' responses, and analyze the test phase. The chapter will conclude by testing our original hypothesis against the user test data and further discussing the results of the testing.

Description of the Participants

Section 3.3 described the type of background information we asked the participants. Below we present the significant characteristics of the group.

The participant group demographics are as follows:
- There were eleven participants in the first round of testing.
- The average participant age is twenty-five years old.
- Two participants were female and nine were male.
- Two participants were left-handed and nine were right-handed.
- Two participants were professionals and nine were students.
- The highest education level completed for four of the participants was bachelor's degree, five had completed master's degrees, and two had had some amount of college education.
- Six of the participants were near-sighted. The others had normal vision. Of the near-sighted group, half used glasses and half used contact lenses as their normal form of vision correction.
- The group all had at least two years experience with IBM/PC systems and UNIX systems. Most participants had some level of experience with Apple Macintosh computers.
Three of the participants had no experience with palmtop computers, four had minimal or days experience with palmtop computers, and four of the population had two months or more experience with palmtop computers.

The entire group had at least one year of experience with Virtual Reality systems.

Many of the group had experience with palmtop operating systems. Six were familiar with PalmOS, four with Windows CE, zero with Microsoft's Pocket PC OS, and four with Apple's Newton OS.

Most participants felt either extremely or very confident about their ability to learn new applications. One individual felt less than moderately confident when learning new applications.

On average the group felt moderately to very comfortable with palmtop computers.

All participants had had a virtual reality experience in the preceding ten days.

All participants had previous experience with IPT systems similar to the C2. Seven participants had experience with head mounted display systems. Nine had experience with home and arcade video game systems.

To summarize, the participant group had a high level of experience with computer systems, virtual reality technology. The relatively low level of palmtop computer experience, high level of education, and high confidence in learning new applications are also significant factors that may affect the subjects performances during the test phases.

Test Round One

We will now discuss the participant's responses to the surveys given as part of the first testing phase. The responses to each question will be described individually, then, a brief analysis will be given. This discussion will also present the experimenter's observations of the participants while executing the tasks.
Description of the Sample

All eleven participants completed the tasks and completed the survey forms successfully. Six participants performed the task in the JAIVE/Builder application first. Five participants began with the VADeT system. On average, the participants required 5.7 minutes orientation time for the VADeT application, and six minutes to complete the task. Orientation time for the JAIVE/Builder application was 7.45 minutes and tasks were completed in an average 10.2 minutes. The experimenter graded all individuals' task completion accuracies with the VADeT application as excellent to above average. The JAIVE/Builder task accuracies were graded as excellent or above average as well.

Experimenter Observations

The experimenter noted the following aspects of the participants' behavior while using the VADeT application:

- Most participants were very mobile while recreating the design template. These participants changed their physical orientation and location several times during the design process.
- One participant placed the VUI Color Panel coplanar with the left wall of the C2. This placement prevented the panel from obscuring the users' view of the design.
- Users did not ask many questions while operating the floating menus, but were occasionally confused about the VUI Color Panel's operation.

The experimenter noted the following while users interacted with the JAIVE/Builder application:

- Users had a difficult time handling both the Qbe device and the C2 wand. Most users placed the non-used device on a table, on a chair, on the floor, or, in the case of the wand, in a pocket.
• Left-handed individuals had additional difficulty selecting a good orientation to hold the Qbe device. This additional difficulty stemmed from the location of the wireless networking card in the upper left hand corner of the device, which disallowed participants to grip that corner of the device with their right hand.
• The hierarchical JAIVE menus confused some participants. They indicated that they could not tell where the operations they wanted to use were located in the hierarchy.
• Participants were not very mobile when carrying the tablet computer.
• Participants tended to do all interactions through the tablet computer even though the task instructions indicated that the wand was available for positioning shapes. Some participants performed all JAIVE operations they could and then switched to the wand device to place the shapes.
• Participants wanted to lock all of the three degree-of-freedom sliders together to perform uniform scale operations. At the time this capability was unavailable.
• Participants felt there were too many cables. The tracker, wand, and tablet computer pen were all wired to different locations around the C2.
• The tablet computer display’s polarization interfered with the shutter glasses and restricted the participants’ views of the JAIVE control panels at certain screen orientations.
• Participants had to forcefully select items from the tablet computer display. The Qbe tablet did not seem to respond well to single click operations.

These experimenter observations will be discussed in further detail in the analysis portion of this section.

Description of Responses

We will now discuss the participants’ responses to the survey questions (Appendix A). Each question is listed followed by the average response, standard deviation, and F-Test value of the responses. The F-
Test value is the one-tailed probability that statistical differences between two groups of data are not significantly different. In this case we have used F-Test to compare the groups who worked with the JAIVE/Builder application against those who worked with the VADeT application first.

A number of the survey questions asked the user to indicate an interaction preference for performing operations in the two test applications. Some questions ask the participant to rate some aspect of the system. A sample question would be "How easy is it to select operations?" Participants answered these questions on a spectrum ranging from 'extremely' to 'not at all'. Participants were asked to place a mark along an axis to indicate their choice. The marker position was assigned a decimal value corresponding to its position. Table 4.1 displays each question, indicates the numerical range for the possible responses, and lists the average value, standard deviation and F-Test value for that response.

Table 4.1 Test Round One Participant Responses to Questions 1-14

<table>
<thead>
<tr>
<th>No</th>
<th>Question</th>
<th>Value Assignments</th>
<th>Avg.</th>
<th>Std. Dev.</th>
<th>F-Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Indicate the method you prefer for creating shapes. Place a mark along the axis that most accurately reflects your preference.</td>
<td>(5) Ray Selected Menu</td>
<td>3.45</td>
<td>1.42</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Either Palmtop Menu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Indicate the method you prefer for selecting shape colors.</td>
<td>(5) Virtual Color Window</td>
<td>1.96</td>
<td>0.59</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Either Palmtop Menu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Indicate the method you prefer for selecting shape textures.</td>
<td>(5) Virtual Menu</td>
<td>2.42</td>
<td>1.03</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Either Palmtop Menu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Indicate the method you prefer for positioning (translating or rotating) shapes given a specific set of coordinates.</td>
<td>(5) Wand Interaction</td>
<td>1.90</td>
<td>1.32</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Either Palmtop Window</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Question</td>
<td>Value Assignments</td>
<td>Avg.</td>
<td>Std. Dev.</td>
<td>F-Test</td>
</tr>
<tr>
<td>----</td>
<td>--------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>5</td>
<td>Indicate the method you prefer for positioning (translating or rotating) shapes at approximate locations.</td>
<td>(5) Wand Interaction (3) Either (1) Palmtop Window</td>
<td>4.81</td>
<td>0.32</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>How well did you understand which palmtop/tablet computer actions you were performing?</td>
<td>(5) Extremely (4) Very (3) Moderately (2) Minimally (1) Not at all</td>
<td>4.24</td>
<td>0.73</td>
<td>0.24</td>
</tr>
<tr>
<td>7</td>
<td>How well did you understand which virtual menu actions you were performing?</td>
<td>(5) Extremely (4) Very (3) Moderately (2) Minimally (1) Not at all</td>
<td>4.25</td>
<td>0.74</td>
<td>0.27</td>
</tr>
<tr>
<td>8</td>
<td>How easy was it to select operations with the palmtop/tablet computer (without regard to the palmtop/tablet computers size, weight, and/or maneuverability)?</td>
<td>(5) Extremely (4) Very (3) Moderately (2) Minimally (1) Not at all</td>
<td>4.10</td>
<td>0.74</td>
<td>0.32</td>
</tr>
<tr>
<td>9</td>
<td>How easy was it to select operations with the virtual menus?</td>
<td>(5) Extremely (4) Very (3) Moderately (2) Minimally (1) Not at all</td>
<td>4.01</td>
<td>0.62</td>
<td>0.47</td>
</tr>
<tr>
<td>10</td>
<td>How intuitive was the palmtop’s operation?</td>
<td>(5) Extremely (4) Very (3) Moderately (2) Minimally (1) Not at all</td>
<td>3.73</td>
<td>0.67</td>
<td>0.96</td>
</tr>
<tr>
<td>11</td>
<td>Was the palmtop’s display quality sufficient for you to read the text and distinguish the controls?</td>
<td>(5) Extremely (4) Very (3) Moderately (2) Minimally (1) Not at all</td>
<td>3.63</td>
<td>1.12</td>
<td>0.59</td>
</tr>
</tbody>
</table>
Table 4.1 (Continued)

<table>
<thead>
<tr>
<th>No</th>
<th>Question</th>
<th>Value Assignments</th>
<th>Avg.</th>
<th>Std. Dev.</th>
<th>F-Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Was your view of the world more obscured by the palmtop interface or by the virtual interfaces (floating menus and windows)?</td>
<td>(5) Palmtop Interface, (3) Equally, (1) Virtual Interface</td>
<td>1.86</td>
<td>1.06</td>
<td>0.24</td>
</tr>
<tr>
<td>13</td>
<td>How comfortable was it to use the tablet computer in the Virtual Environment. Please consider only the physical aspects of interaction (weight, device size, maneuverability etc.)?</td>
<td>(5) Extremely, (4) Very, (3) Moderately, (2) Minimally, (1) Not at all</td>
<td>2.29</td>
<td>0.68</td>
<td>0.23</td>
</tr>
<tr>
<td>14</td>
<td>If a tablet computer of ideal size and ergonomic design were available for daily interactions, would you find it useful?</td>
<td>(5) Extremely, (4) Very, (3) Moderately, (2) Minimally, (1) Not at all</td>
<td>4.04</td>
<td>0.83</td>
<td>0.57</td>
</tr>
</tbody>
</table>

Question 15 asked users to grade each palmtop interface panel based on how beneficial they felt it would be in virtual environment applications. The grading scale and scores ranged from very beneficial (5) to not beneficial at all (1). The participants' average responses and standard deviation are ranked by average value in Table 4.2.

Participants were encouraged to make written comments on their answers for questions 13 and 14. Question 13 asked how comfortable it was to use the JAIVE/Palmtop device. Participants were encouraged to elaborate on which aspects of the tablet computer were comfortable or uncomfortable. Their responses have been condensed into the following main points, which are:

...
Table 4.2 Test Round One Participant Responses to Question 15

<table>
<thead>
<tr>
<th>Panel Name</th>
<th>Average Value</th>
<th>Standard Deviation</th>
<th>F-Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Adjustment</td>
<td>4.5</td>
<td>0.84</td>
<td>0.13</td>
</tr>
<tr>
<td>Button/Operation Panel</td>
<td>4.3</td>
<td>0.26</td>
<td>0.76</td>
</tr>
<tr>
<td>Information/Help Window</td>
<td>4.2</td>
<td>0.98</td>
<td>0.27</td>
</tr>
<tr>
<td>Map/navigation Window</td>
<td>4.0</td>
<td>1.10</td>
<td>0.55</td>
</tr>
<tr>
<td>Annotation/Note</td>
<td>3.8</td>
<td>1.17</td>
<td>Unable to Compute</td>
</tr>
<tr>
<td>Color Selector</td>
<td>3.8</td>
<td>0.98</td>
<td>0.27</td>
</tr>
<tr>
<td>Picture Viewer</td>
<td>3.3</td>
<td>1.63</td>
<td>0.78</td>
</tr>
<tr>
<td>Three degree-of-freedom</td>
<td>3.3</td>
<td>1.03</td>
<td>0.87</td>
</tr>
</tbody>
</table>

- The Qbe device does not comfortably fit in one hand. The Qbe computer is too heavy and too large for long-term use. Further, holding the Qbe-tablet restricts motion.
- Users indicated they would like a strap or harness for Qbe computer. Others indicated a preference for a palmtop device that attached to the arm.
- Users had to support the Qbe with their body and 'stoop' over to see display. As a result, users were worried that the glasses would fall off when user tilted head down to look at the display.
- The Qbe computer failed to recognize many single-click operations.
- Users indicated a preference for a non-tethered pen.
- Users felt the entire system used too many wires. Users suggested using a smaller wand with lighter cabling.

Question 14 asked participants, if an ideal Palmtop device were available for daily use, how useful would it (the JAIVE system) be? Their responses included the following:

- The JAIVE system would be very useful for selecting large icons from menus and adjusting parameters.
- The JAIVE system would be useful for making short notes, but not for typing large amounts of data.
• The connection to the Internet would be useful for viewing help or product information.

• JAIVE would work well when accurate numerical data manipulation was necessary.

• Many traditional Graphical User Interfaces (GUI's) could be implemented quickly with the JAIVE system.

• Participants suspected that novice VR users would more easily understand the JAIVE windows than 'custom' interfaces.

• Participants' see good potential for instantaneous navigation via the map view panel.

• One participant noted that when interacting with the tablet computer they focused exclusively on the JAIVE system and not the virtual environment.

The survey also invited participants to suggest additional interaction methods they felt should be implemented in the JAIVE system. They suggested the following:

• JAIVE could provide file management operations from within the virtual environment.

• JAIVE could present a single page hierarchy view featuring all of the interface controls.

• JAIVE could be useful for application specific interactions.

• JAIVE could be used for switching between applications.

• JAIVE could provide a 2D sketchpad, which could then send sketched data into the 3D virtual environment.

• Video playback or video conferencing would be good possible uses for the system.

• JAIVE could display application information including performance statistics or author information.

• It would be useful to integrate the map and three degree-of-freedom panels together with user position and orientation information for a robust navigation system.

Finally, participants were asked to write any additional comments they may have had. The participants' comments include:

• Users suggest glove interactions with the JAIVE system as an alternative to handling the wand.
• Participants suggested providing a holder for the wand and/or palmtop computer when not in use.
• Users requested higher precision and quicker response from the three degree-of-freedom panels.
• Users want the ability to adjust a JAIVE slider without looking directly at the palmtop display.
• Some users indicated a preference to have the tablet computer off to the side of the virtual environment where they could pick up the device only when needed.
• Users noted that switching between the 2D and 3D views is somewhat distracting. It takes them a second to adjust to the brightness difference between JAIVE and the C2 environment.

Test Phase One Analysis

The first phase of testing has yielded many important pieces of information. We divide this information into the categories of user preference, control panel effectiveness, palmtop device characteristics, and areas for improvement.

In examining the user preference related questions, we see that users slightly preferred the ray-selected menus above JAIVE system for creating shapes. Users moderately preferred the palmtop device for selecting shape colors and performing shape transformations to specific coordinates. JAIVE was also preferred slightly for selecting shape textures. Wand interaction methods were strongly favored for approximate shape placement. Users understood the JAIVE operation selections equally as well as they did the wand/virtual menu selections. Users also found it equally as easy to select operations with the JAIVE panel as they did the virtual menus. Users found the JAIVE computer to be moderately to very intuitive to operate. Users indicated that the virtual interface windows obscured their view significantly more than JAIVE computer. Users indicated that, given an ideal hardware device, the JAIVE system would be very useful for VR interactions.

In gauging the potential benefit of the interaction panels, users saw the most benefit in the parameter adjustment, button operations,
and help/information panels. Users saw the least potential benefit from the picture viewer and the three degree-of-freedom panels. All of the panels seemed to fare well overall but there was a high amount of deviation in the scores of a number of the panels.

In examining the palmtop device characteristics, participants indicated that the display quality was moderately to very sufficient to distinguish the controls even with the display polarization issue. They indicated that the device was moderately to minimally comfortable to use and cited size, weight, and polarization as the chief detractors from the device’s comfort.

The chief areas for improvement we’ve identified from this round of testing include the JAIVE hardware platform and slider accuracy. We clearly need to select a smaller, more lightweight palmtop computer. The computer display should be completely non-polarized, with a non-tethered pen, and the user should be able to carry the device more easily. The device should also incorporate dedicated soft keys for increasing or decreasing values. This would allow users to adjust JAIVE panel sliders without focusing on the device. The three degree-of-freedom sliders definitely need to be changed to accommodate floating-point data entry as opposed to integer values. We should also consider using a smaller wand or glove as the IPT system interaction device.

Users suggested a number of potential additional interactions for the system. We will not be adding the new interface types for the second round of testing. Instead we will mend the shortcomings in the current system and ask the users more specific questions about the system’s behavior and usability.

**Test Round Two**

We now discuss the second testing phase. As described in Section 3.3, the second test round asked participants to recreate the same design template as given in the first round. This time participants were given exact information about the shapes’ positions, rotations, and scales. Participants used the Intermec computer for this test round and were encouraged to use the tablet’s soft keys to input numerical
values. Participants were also discouraged from using the wand during this exercise.

We now discuss the participant's responses to the surveys given as part of the second testing phase. The responses to each question will be described individually then a brief analysis will be given.

Description of the Sample

Eight individuals participated in the second round of testing. Originally we had hoped to retest all eleven first round participants in the second round however, a number of the participants were unavailable to retest. The second test group consisted of six first round participants and two more individuals with similar experience characteristics as the first round subjects.

All participants were able to complete the tasks successfully. All subjects displayed either excellent or above average accuracy in recreating the design template.

Experimenter Observations

The experimenter did not note many marked changes in participants' behaviors when compared to last round. One key difference was that participants were somewhat more mobile with the Intermec computer than they had been with the Qbe device used in the first round. Another difference was that subjects did not have to spend any time or effort changing between the wand and the tablet computer.

Also, most users seemed to have an easier time selecting items from the Intermec computer than the Qbe. Some of this difference is attributed to the smaller size of the Intermec pen. Also, the Intermec touch screen seemed much more responsive and accurate than the Qbe screen. Even so, a couple of users very forcefully selected screen items.
Description of the Responses

The participants' responses to the second round survey questions (Appendix A) are discussed below. The questions were similar to those from the first round of testing. Participants were presented with seven questions and asked to place a mark along an axis corresponding to their response to the question. Table 4.3 presents the results of the second round questionnaire. Each question is listed with the value assignments, average value, and standard deviation of the responses. The second test round only required participants to operate one of the applications so F-Test analysis was not performed on this data set.

Users were asked to elaborate on their responses to a number of the survey questions. When asked how comfortable was it to use the tablet computer in virtual environment users responded as follows:

- Users felt the Intermec device was much more comfortable than the Qbe device.
- Users reported that the Intermec device weight and size were nice, but there was no good way to hold on to the device.
- The placement of the network card along the left side of the Intermec computer made it difficult for users to hold the device from that side.
- Users were concerned that the shutter glasses slid off of their head when they looked down at the display.

When asked to write any additional comments they had about the system users wrote the following:

- Users suggested that the tablet computer soft keys should be backlit to aid their selections.
- Users suggested that the tablet computer brightness be matched to the C6/C2 projectors to help their eyes adjust to the various displays.
- Users noted that they had to switch between the information help panel and the various three degree-of-freedom panels several times to complete their task. They would have preferred to have the information help panel available at all times.
• Some users noted that when asked to enter accurate dimensions they used the tablet computer soft keys almost exclusively.
• One user noted that it would be nice if the three degree-of-freedom sliders moved along discrete intervals.

Table 4.3 Test Round Two Participant Responses to Questions 1-7

<table>
<thead>
<tr>
<th>No</th>
<th>Question</th>
<th>Value Assignments</th>
<th>Avg.</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>How well did you understand which palmtop/tablet computer actions you were performing?</td>
<td>(5) Extremely</td>
<td>4.81</td>
<td>.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>How easy was it to enter scale, position, and rotation information with the palmtop/tablet computer using the value sliders?</td>
<td>(5) Extremely</td>
<td>3.69</td>
<td>.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>How easy was it to enter scale, position, and rotation information with the palmtop/tablet computer using the tablet computer soft keys?</td>
<td>(5) Extremely</td>
<td>4.79</td>
<td>.36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Was the palmtop’s display quality sufficient for you to read the text and distinguish the controls?</td>
<td>(5) Extremely</td>
<td>4.31</td>
<td>.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>How comfortable was it to use the table computer in the Virtual Environment. Please consider only the physical aspects of interaction (weight, device size, maneuverability etc.)?</td>
<td>(5) Extremely</td>
<td>4.06</td>
<td>.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>How useful was the information window in providing shape dimensions for this task?</td>
<td>(5) Extremely</td>
<td>4.68</td>
<td>.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>If a tablet computer of ideal size and ergonomic design were available for daily interactions would you find it useful?</td>
<td>(5) Extremely</td>
<td>4.60</td>
<td>.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The next section offers an analysis of these responses. Before that discussion, we compare the results of questions that were similar between the first and second test rounds. The questions and average response values are listed in Table 4.4.

Table 4.4 Comparisons of First and Second Round Answers

<table>
<thead>
<tr>
<th>Question Number</th>
<th>Question</th>
<th>Value Assignments</th>
<th>Round One Avg.</th>
<th>Round Two Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round One</td>
<td>Round Two</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>How well did you understand which palmtop/tablet computer actions you were performing?</td>
<td>(5) Extremely</td>
<td>4.23</td>
<td>4.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Was the palmtop's display quality sufficient for you to read the text and distinguish the controls?</td>
<td>(5) Extremely</td>
<td>3.63</td>
<td>4.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>How comfortable was it to use the table computer in the Virtual Environment. Please consider only the physical aspects of interaction (weight, device size, maneuverability etc.)?</td>
<td>(5) Extremely</td>
<td>2.29</td>
<td>4.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>If a tablet computer of ideal size and ergonomic design were available for daily interactions would you find it useful?</td>
<td>(5) Extremely</td>
<td>4.03</td>
<td>4.60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) Moderately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) Not at all</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This section has presented the data gathered during the second user test period. The next section analyzes this data to determine the impact of our JAIVE system changes on user responses.

Test Phase Two Analysis

As mentioned in chapter three, some changes were made to the JAIVE system based on the first round of user testing. The second round data indicates that most of the changes have improved the usability of our system. Here we analyze the impact of our changes according to the second round responses. The second round survey form is relatively short and a number of the questions are identical those of the first round. As such, we will examine each question individually below and when appropriate, compare the responses to those of the first test round.

In the first question, users reported that they understood their palmtop/tablet computer operations extremely to very well. Also, the relatively low deviation in response seems to indicate good agreement amongst all participants over this point. Interestingly, users' average responses to this question were 0.6 higher (on a five point scale) than the first round (Fig 4.4). The response deviation also decreased significantly from the first to the second round, decreasing from .73 to .35 in the second round.

Users responses also improved by 0.6 for question seven, which asked users to rate the usefulness of a tablet computer of ideal ergonomic design in daily VR interactions. In the first round, the average response indicated very useful, in the second, halfway between very and extremely useful. Also, the response deviation for this question reduced in half from the first to the second round. The first round deviation was 0.83 and in the second round was 0.43. We suspect that the ergonomic differences between the Qbe and Intermec computers were significant factors in this improvement.

When asked how comfortable the palmtop/tablet device was in the virtual environment (question five), users indicated that it was very comfortable. They also indicated that the tablet computer's resolution was very sufficient to read the text and distinguish the controls
(question four). While the Qbe computer was rated as minimally comfortable in round one, the Intermec was rated very comfortable in round two. The deviation in responses was approximately .8 for both rounds. This indicates that the Intermec computer is much more highly preferred for performing tasks in the virtual environment.

The second and third questions asked users how easy it was to input shape information by positioning the three degree-of-freedom sliders and entering data into the three degree-of-freedom text input fields with the tablet computer's soft keys. Users found the sliders to be very to moderately easy to use with an average and deviation of 3.7 and .90 respectively on a five-point scale. Users found the Intermec soft keys to be extremely to very easy to use with an average of 4.79.

The second and third questions indicate that the soft keys are quite helpful to users when performing exact object positioning. The sliders proved less desirable in performing these tasks. This may be related to the resolution of the slider manipulation. First round participants indicated that the sliders should be modified to accommodate floating-point data entry. As a response, the sliders were changed to allow two decimal places of granularity. The second round responses seem to indicate that there is too much granularity in some cases, which makes it difficult for users to indicate integer responses. The slider precision issue is unresolved at this point.

User responses also indicate that users would prefer to press the 'enter' soft key instead of using the three degree-of-freedom set buttons to indicate that a variable change is complete. This change is small and could be made such that both methods are available.

Question number six asks users how useful the information window is in providing shape dimensions for their use in the task. Users responded that the panel was between very and extremely useful for this purpose. This score is a bit higher than users first round estimated potential benefit for the information/help panel. The author suspects that using this panel to accomplish a task has improved participants understanding of its usefulness. If so, this fact should encourage us to retest users with the other low scoring interaction panels once they have been properly integrated into a mature VR application.
Discussion

The first and second testing rounds have yielded some very positive results regarding our palmtop interaction system. Users have indicated that the system is at least as understandable as more commonly used virtual menu and floating icon interface methods. Users also indicated that the system is fairly easy to operate and obscures the virtual world less than the virtual interface methods we tested.

We have also found that users prefer the palmtop device for some interactions, but not all of them. For example, users prefer palmtop methods for entering specific information such as shape coordinates, scales, and rotations. However, when approximate placement is sufficient, users prefer direct manipulation methods.

One major issue we have noticed concerns the type of tablet/palmtop computer that the JAIVE system runs on. The ergonomics of the device seem to have an impact on the user's mobility and estimation of the system's usability. Also, the inclusion of physical numerical soft keys into the device greatly assists users when entering specific shape coordinates. The benefit of tablet computer soft keys was underestimated initially. That benefit is now considered a significant factor when selecting the tablet/palmtop computer. Perhaps future device selections should incorporate entire keyboards.

Using the data from the first two rounds of testing we have seen that there can be benefits to performing some virtual environment interactions using the palmtop interface. We have seen the impact that the palmtop computer's design can have on the user's experience with the system. The next chapter takes the analysis of this data a step further. In chapter five we suggest which types of interactions and applications palmtop devices should be considered for.
CHAPTER FIVE

CONCLUSIONS

In this chapter we will summarize the efforts made during this thesis project. The discussion will review the palmtop interaction device effectiveness data presented in Chapter 4. From this data we shall discuss which interaction methods we have found to be more, equally, or less effective when performed through a palmtop interaction device. Subsequently we will suggest which VR applications can potentially be improved by integrating palmtop interaction methods. We will suggest future areas of research in the area of palmtop or tablet computer interaction. We will then conclude this thesis with a brief review of its contents and a list of the contributions that this effort has made to the VR community.

Palmtop Device Interaction Effectiveness

The user test results described Chapter 4 seem to indicate that our palmtop interaction system is at least as understandable and easy to use as comparable virtual user interface methods for performing operations or selecting colors. The system also proved to be moderately intuitive for users and relatively easy to learn. Users seemed to feel that the system held a great amount of potential for improving certain types of interactions and less potential to improve others. At this point, based on our user test data, we categorize the palmtop interaction methods we have tested into the categories of more, less, and similarly effective in virtual environment interactions.

More Effective Methods

The list of more effective interaction methods includes exact three degree-of-freedom geometry manipulations, operation selection, color selection, and parameter adjustment. User test participants survey form responses indicated a clear preference toward the palmtop device for performing these types of interactions. Users estimated that the palmtop device would be well suited for adjusting parameters,
performing push button style operations, or for viewing help information.

Interestingly the three degree-of-freedom interaction window received the lowest estimated benefit value of all the panels, but users clearly preferred the method for exact shape manipulation when given specific values. The highest potential benefit was given to the parameter adjustment panel.

Similarly Effective Methods

The similarly effective interaction methods are ones for which we cannot yet demonstrate a tremendous advantage for either the virtual or palmtop methods. These interaction methods include using the palmtop computer as a map or navigation aid, for selecting colors, or for creating annotations.

Users indicated a preference toward the palmtop device for selecting colors, however, the color panel’s estimated benefit values were relatively low compared to the group of interaction panels. We attribute part of this to the somewhat awkward integration of the virtual color window into the VADeT application.

We cannot make any definite conclusion regarding annotation operations. User test participants were asked to study the appearance and operation of the annotation panel, but were not required to use it. Users’ written comments indicated that they would prefer a physical keyboard for inputting alphanumerical information into the annotation panel as opposed to the text recognition or simulated keyboard methods of the palmtop computers. It is believed that this preference has had an affect on users’ estimations of the palmtop’s potential benefit in creating annotations.

Less Effective Methods

Less effective interaction methods include approximate shape positioning and possibly picture viewing. Users clearly preferred direct manipulation methods for positioning virtual objects in approximate locations. Users believed that the picture viewer held low
potential for improving VR applications. Part of this may be attributed to the lack of a clear example of how this interaction method would be used in a virtual reality application.

This section has indicated, based on our preliminary findings, which specific palmtop interaction types seem to be more, less, or similarly effective. Given these rough estimates, we will now try to suggest applications that stand to benefit the most from palmtop interaction methods.

Suggested Applications of Palmtop Interaction Device

Palmtop interaction methods seem to have the most potential benefit for use in applications that require a large number of precise parameter adjustments. Example applications or tasks might include computational steering, data entry, information retrieval, or virtual design tools.

The parameter adjustment window could be quite helpful when interacting with real-time simulations of real-world systems. The parameter adjustment ability allows users to select and modify variables with as much resolution as the application requires. A palmtop computer with alphabetical or numeric keys is ideal for entering and modifying numerical data. The Internet capabilities of palmtop devices could potentially be used whenever information or web resources related to the virtual environment simulation are available. An example of Internet integration might include a virtual factory monitoring application in which the status of all assembly stations or parts can be tracked via the net. This factory monitoring tool could be used to visualize bottlenecks and adjust factory operations in real-time.

It seems that the tablet computer can be useful for some three-dimensional modeling applications. It has been demonstrated to be preferable for certain virtual environment interactions in computer aided design applications. In addition to the features discussed above, it could also be useful for allowing a user to navigate through an object selection hierarchy. We believe there are many more potential
uses for this technology. The next section on future research describes additional possibilities for palmtop interaction.

Future Research

There are several areas of future work for palmtop interaction systems for immersive environments. This research could involve improving the palmtop computer, increasing the number and type of interactions, including intelligent or adaptive user interface methods, increasing the systems flexibility, integrating distributed interactions, or more. We discuss each area below.

The palmtop computing devices we have used are being improved everyday. The miniaturization of system components has lead to increased computation power. This increase has allowed us to use more complex interaction methods and provide a more responsive system to the user. Another effect has been to reduce the size and weight of the device, which allows the user to work with device for longer periods of time.

Additional interaction methods could be developed for the palmtop computer. The interactions presented in this effort cover a wide range of tasks and application areas. The user study indicated that users would like to have file-browsing capabilities through the tablet computer. Another improvement might be to add spatial awareness techniques to the palmtop interface. A tracked palmtop computer could be used to scan the virtual space similarly to a Star Trek®™ tricorder. The interface could then reconfigure itself as appropriate for the object the user has focused on. The tablet computer's multimedia capabilities might allow it to be used for audio/video streaming or for speech recognition interactions. The tablet computer could also be used as a two-dimensional sketchpad, which would then transmit sketched data into the virtual environment.

Intelligent and Adaptive User Interface methods could also be integrated into the palmtop interaction system. Adaptive interfaces were described in Chapter 2 of this thesis. These methods have not been included in the JAIVE system design at this time. However, if used properly, adaptive interface methods could potentially improve the
user's experience in the virtual environment. The interface could study the user's actions over time and make predictions about the user's future actions. These predictions could then be used to automate palmtop user interface selections or to distinguish visually the more or less relevant interaction controls.

The JAIVE system could be modified to increase the system's overall flexibility. This flexibility could allow users to create their own interaction panels for custom applications. While this capability exists in the system to some extent, it could be taken even further by providing users with a thin JAIVE network service client. This thin client would allow the user to easily integrate JAIVE communications into their C, C++, or JAVA application.

This system could incorporate distributed application methods to facilitate multi-user environments. As mentioned earlier, audio/video streaming could serve to connect users with their counterparts in distributed worlds. Users could use the palmtop computer to instantly locate and travel to other the user’s locations. The palmtop could also provide a wealth of information about the status of other users in the virtual space.

This section has listed a number of potential palmtop research areas. The topics listed above were all hardware or software related. There is also the potential for interdisciplinary studies to occur. Research could also be done to examine the psychological impact of palmtop computers on VR interactions. A more specific question might be to ask how the separation of the user interface from the graphics affects the user's sense of immersion. Another interesting question is whether the palmtop computer simplifies a new VR system user's performance with applications. It may also be interesting to examine the effect that training with the palmtop computer interface on a workstation would have on performance in the virtual environment and vice versa. With these questions, we end our discussion of future research areas. We hope that more individuals take interest in this interaction method and pursue the effort.
Final Summary

This thesis has addressed a number of topics related to the design, evaluation, integration, and use of palmtop interaction devices in Immersive Virtual Environments. We will now overview the contents of the thesis and summarize the contributions of this research effort.

The first chapter introduced the topic of palmtop computers for VR systems. The research questions we asked concerned whether or not palmtop devices could strengthen deficient interaction methods and what kind of interactions could be improved through the use of a palmtop device. We stated that this project would survey background material in the areas of Interaction, Usability, and Intelligent User Interfaces. This project would also design an interaction model for palmtop interaction and implement this model in a prototype system. It was stated that this project would perform experimentation to determine the effectiveness of the prototype system. Finally, the project would identify future areas of research for palmtop interaction.

Chapter Two presented detailed information about theories of interaction and interaction in Immersive and Non-immersive systems. We addressed a number of current interaction methods and described some of the strengths and weaknesses of those methods. Chapter Two also discussed the field of Usability. We examined a number of Usability design methods and evaluation techniques. Along with discussion, we looked into some of the common mistakes made when designing for usability. The balance of Chapter Two introduced the area of Intelligent User Interfaces. This information was presented to indicate potential future directions for palmtop interaction systems even though the majority of the material was not directly applied in this research effort.

Chapter Three presents our concept for a palmtop interaction system. This chapter discusses the Java based Interface to the Virtual Environment (JAIVE). In developing this system we considered the Interaction and Usability information surveyed during the literature search. We examined a number of related interaction systems. We identified the types of interactions we wished to implement in the system and presented an overview of the system's design. This chapter
also describes the user test experimentation we wished to perform in order to determine whether these interaction methods were more or less effective. We described two rounds of user testing, both involving the use of the palmtop device in virtual primitive modeling applications. We also identified the survey tools that were used to gather feedback information from users about their testing experience.

Chapter Four discusses the results of the user testing. We describe the participant group in terms of their familiarity with the technologies related to this project. We summarized their survey responses and analyzed the quantifiable responses from both rounds of testing. We then discussed the results of the user testing rounds which lead to a few JAIVE control panel implementation changes and motivated us to examine other palmtop devices for a more appropriate fit with the JAIVE system.

Chapter Five offers a conclusion to this effort. Here we have analyzed the effectiveness of the various palmtop interaction methods. We have suggested which VR applications have the greatest potential for improvement through the use of palmtop computers. We also presented a number of areas for future research into this area.

In conclusion we believe that this research effort has contributed to the fields of virtual reality and human computer interaction by:

- Identifying a potential area for the use of palmtop computers in virtual reality applications.
- Presenting an updated vision for palmtop interaction in VR applications by harnessing the power of present day computing power, display quality, wireless networking, and cross platform software tools.
- Creating a tool for developers to quickly and easily integrate a number of palmtop interaction methods into their applications.
- Presenting experimental data that gives a preliminary indication as to how effective or ineffective certain palmtop interaction methods are with this system.
- Suggesting applications of this technology and identifying a number of potential future research areas.
With this listing of contributions we conclude this thesis. We hope that this effort will inspire new thought into immersive environment interactions and encourage the development of new interaction techniques.
APPENDIX
USER TEST MATERIALS

Introductory Survey

Basic Information

Age: __________________________
Date: __________________________

Gender (circle one):

Female    Male

Dominant Hand (circle one):

Left    Right

Occupational Information:

______ Student
______ Faculty
______ Professional

Subject ID #: __________

Educational Background (indicate highest level only):

______ High School
______ Some College
______ 2 Year Degree
______ Bachelors or 4 year
______ Masters
______ Ph.D.

Major/Minors (if applicable):

________________________________________

Vision Information:

Are you?

______ Near Sighted (Can see things that are close without vision correction)
______ Far Sighted (Can see things that are far)
______ Normal Vision

At the time of the study will you be wearing either of the following?

______ Glasses
______ Contacts

Is this your normal means of vision correction (circle one)?  YES  NO

Language:

Are you a fluent in English (circle one)?  Yes  No
Previous Experiences

Subject ID #_____

Indicate (roughly) the number of days/months/years experience you have with:

- IBM or compatible computer: ________ days months years
- Apple Macintosh: ________ days months years
- UNIX systems: ________ days months years
- Palmtop/Tablet Computers: ________ days months years
- Virtual Reality: ________ days months years

Please indicate which Palmtop/Tablet operating systems you have used. (Select all that apply).

- PalmOS
- Windows CE
- Pocket PC
- Newton
- Other: (please specify)__________

How comfortable do you feel about learning new computer applications? Please indicate on the scale below.

Extremely Very Moderately Minimally Not at all

How comfortable you feel with palmtop computers? Please indicate on the scale below.

Extremely Very Moderately Minimally Not at all

How long ago was your most recent virtual reality experience?

_______ days months years

Please indicate which Virtual Reality systems you have used. (Select all that apply).

- Immersive Projection System (CAVE/C2/C6/SSVR)
- Enclosed Head Mounted Display
- Augmented Reality System
- Power wall
- Stereo Monitor
- Home Video Games
- Arcade Video Games
First Round
Participant Response

Subject ID#: 

Please compete this survey as accurately as possible. Your answers will remain confidential.

1. Indicate the method you prefer for creating shapes. Place a mark along the axis that most accurately reflects your preference.

| Ray Selected Menu | Either Palmtop Menu |

2. Indicate the method you prefer for selecting shape colors.

| Virtual Color Window | Either Palmtop Window |

3. Indicate the method you prefer for selecting shape textures.

| Virtual Menu | Either Palmtop Menu |

4. Indicate the method you prefer for positioning (translating or rotating) shapes given a specific set of coordinates.

| Wand Interaction | Either Palmtop Window |

5. Indicate the method you prefer for positioning (translating or rotating) shapes at approximate locations.

| Wand Interaction | Either Palmtop Window |

6. How well did you understand which palmtop/tablet computer actions you were performing?

| Extremely | Very | Moderately | Minimally | Not at all |

7. How well did you understand which virtual menu actions you were performing?

| Extremely | Very | Moderately | Minimally | Not at all |
8. How easy was it to select operations with the palmtop/tablet computer (without regard to the palmtop/tablet computers size, weight, and/or maneuverability)?

Extremely | Very | Moderately | Minimally | Not at all

9. How easy was it to select operations with the virtual menus?

Extremely | Very | Moderately | Minimally | Not at all

10. How intuitive was the palmtop’s operation?

Extremely | Very | Moderately | Minimally | Not at all

11. Was the palmtop’s display quality sufficient for you to read the text and distinguish the controls?

Extremely | Very | Moderately | Minimally | Not at all

12. Was your view of the world more obscured by the palmtop interface or by the virtual interfaces (floating menus and windows)?

Palmtop Interface | Equally | Virtual Interface

13. How comfortable was it to use the tablet computer in the Virtual Environment. Please consider only the physical aspects of interaction (weight, device size, maneuverability etc.)?

Extremely | Very | Moderately | Minimally | Not at all

Please elaborate on the Question 13. What aspects of the tablet computer were comfortable or uncomfortable?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
14. If a tablet computer of ideal size and ergonomic design were available for daily interactions, would you find it useful?

| Extremely | Very | Moderately | Minimally | Not at all |

Please elaborate on Question 14. Use the back of the paper if you need additional space.

15. Please grade each palmtop interface panel based on how beneficial you feel they would be in virtual environment applications (5 = very beneficial, 3 = moderately beneficial, 1 = not beneficial at all). Reference the color printout of the interaction panels if you cannot remember a panel’s name.

- 3 Degree of Freedom Manipulation (Translate, Rotate, Scale, and/or Navigation)
- Annotation/Note Window
- Button/Operation Window
- Information/Help Window
- Map/Navigation Window
- Picture Viewer
- Color Selector
- Parameter Adjustment Window

Please suggest any additional interactions that you feel the palmtop device could be useful for. Use the back of the paper if you need additional space.

Please write any additional comments you may have here.

© Thank-You for participating in this experiment ©!
Second Round
Participant Response

Subject ID#: __________

Please complete this survey as accurately as possible. Your answers will remain confidential.

1. How well did you understand which palmtop/tablet computer actions you were performing?

| Extremely | Very | Moderately | Minimally | Not at all |

2. How easy was it to enter scale, position, and rotation information with the palmtop/tablet computer using the value sliders?

| Extremely | Very | Moderately | Minimally | Not at all |

3. How easy was it to enter scale, position, and rotation information with the palmtop/tablet computer using the tablet computer soft keys?

| Extremely | Very | Moderately | Minimally | Not at all |

4. Was the palmtop's display quality sufficient for you to read the text and distinguish the controls?

| Extremely | Very | Moderately | Minimally | Not at all |

5. How comfortable was it to use the tablet computer in the Virtual Environment. Please consider only the physical aspects of interaction (weight, device size, maneuverability etc.)?

| Extremely | Very | Moderately | Minimally | Not at all |

Please elaborate on the Question 5. What aspects of the tablet computer were comfortable or uncomfortable?

__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
6. How useful was the information window in providing shape dimensions for this task?

[ ] Extremely    [ ] Very    [ ] Moderately    [ ] Minimally    [ ] Not at all

7. If a tablet computer of ideal size and ergonomic design were available for daily interactions, would you find it useful?

[ ] Extremely    [ ] Very    [ ] Moderately    [ ] Minimally    [ ] Not at all

Please write any additional comments you may have here.

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

!© Thank-You for participating in this experiment AGAIN ©!
REFERENCES


http://www.cs.cmu.edu/~sage

http://www.sensable.com


