DESIGN AND EVALUATION OF POKESPACE: A BIMANUAL HAPTIC INTERACTION TECHNIQUE

by

Thomas N. Smyth
B. Sc., Memorial University of Newfoundland, 2003

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Summer 2006

APPROVAL

Name:	Thomas N. Smyth
Degree:	Master of Science
Title of thesis:	Design and Evaluation of Pokespace: A Bimanual Haptic Interaction Technique

Examining Committee: Dr. Ze-Nian Li
Chair

Dr. Arthur Kirkpatrick, Senior Supervisor

Dr. Daniel Weiskopf, Supervisor

Dr. Karon Maclean, External Examiner,
Associate Professor of Computer Science,
University of British Columbia

Date Approved: August 8, 2006

SIMON FRASER UNIVERSITY library

DECLARATION OF PARTIAL COPYRIGHT LICENCE

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection, and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada

Abstract

This thesis presents a new approach to haptic interaction technique design in which haptic feedback is displayed with a device held in the non-dominant hand, while the dominant hand controls a standard mouse. I believe that this approach has the potential to increase the fluency of everyday human-computer interaction by enabling a more effective division of tasks between the haptic and visual modalities. These ideas are expounded in a set of principles intended to guide the design of such techniques. I also present Pokespace, a novel interaction technique which follows those principles. Finally, I describe a series of three user studies intended to investigate and evaluate both the design principles and Pokespace. The results of the studies, though not unanimously positive, confirmed that Pokespace has the potential to support interaction without visual attention, and suggested several improvements to both the interaction technique and the underlying principles.

Keywords haptic; human computer interaction; interaction technique; attention

Acknowledgments

My introduction to SFU in the summer of 2004 could not have been more pleasant largely thanks to Dr. Binay Bhattacharya, who helped tremendously to make me feel at home in a foreign land, and to introduce me to the world of research.

My work was financially supported by an NSERC PGS-M scholarship and several SFU scholarships, for which I am very grateful. I also thank Stella Atkins for use of the Tobii eye tracker.

The dependably amicable environment of my home on McSpadden Avenue in Vancouver has made my time at SFU ever more enjoyable. For that I must thank my dear housemates, especially Laura Soch, who has been there for me from the start.

My research has often depended upon volunteer participation in pilot user studies, which my friends were always ready to provide. In particular, I thank Steven Bergner, Jonathan Kavanagh, and Dylan Tisdall for their participation and insight.

The preparation of this thesis was generously supported by the School of Computing Science administrative staff, especially Heather Muckart and Val Galat. I thank them for their patience, organization, and generous smiles.

Throughout the most intensive stages of my thesis work, I was very fortunate to be supported by my beautiful and talented girlfriend Catherine Andersen, whose door and heart were always open, no matter how late the night's last #135 bus delivered me to them.

Before and since they bought me my first computer, my parents James and Sharman have been nothing but supportive of me, not only of my interest in computing, but of most every other interest I have had the inclination to adopt. For such unwavering trust, love, and encouragement, I remain forever in their debt.

For their insightful comments and helpful advice, I am grateful to my committee members Dr. Daniel Weiskopf and Dr. Karon MacLean. Their confirmation of the validity of

my work was essential to my confidence in it.

Finally, I offer my sincere thanks to my senior supervisor Dr. Ted Kirkpatrick for the wisdom and grace with which he has guided me over the past two years. Through the gamut of my self-doubt, indecision, and frustration, he has supplied me with carefully measured amounts of direction and autonomy. The sound of his rising from his office chair to cheerfully answer my knock, no matter what his engagement, will stay with me always, as will the many invaluable lessons he has taught me in diverse matters of research, knowledge, and life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Our sense of touch, also known as our *haptic* sense, is extremely important for our interaction with the world around us. It constantly monitors the entire surface of our bodies, and helps us to identify and explore objects. It also plays a key role in our remarkable ability to precisely and fluently manipulate objects with our limbs.

One need look no further than the three basic needs of human life to demonstrate the importance of touch to our existence. Preparation of food, from the peeling of an orange to the perfection of a soufflé, depends upon fine motor movements which are inextricably controlled using feedback from our sense of touch. Our haptic sense allows us to perceive features of objects that cannot be seen due to occlusion, darkness, or lack of visual attention. This ability is essential to intricate manipulations like knitting or sewing, where occlusion is unavoidable. And the construction of an adequate shelter requires the lifting of heavy objects, when our haptic sense is used to gauge how much force is required to do the job.

Not only can we readily accomplish each of these tasks, we do so with a grace and fluidity which is unique to our species, and which has allowed us to move past such basic tasks and on to more still more intricate ones. The modern desktop computer is a notable example of the culmination of that process of specialization. It is an enormously powerful and intricate multipurpose tool. However, it makes surprisingly little use of our sense of touch.

To be fair, the two primary devices we use to interact with computers, the keyboard and mouse, do offer a substantial haptic experience. Both the buttons on a mouse and the keys on a keyboard produce a satisfying haptic effect when pressed, and the edges and raised bumps on keyboard keys help us keep our fingers on the right keys without looking. But this feedback suffers from fundamental limitations — it is mechanical, passive, and totally divorced from the internal state of the system. It is thus quite limited in comparison to the complexity and variability of the tasks performed using computers.

This seems a shame. For other sensory modalities, notably vision and audition, computers and other digital devices have typically offered much *more* dynamic feedback than their analog counterparts, with great benefit. For example, computer displays allow data to be visualized and manipulated much more easily than paper or acetate ever could. Similarly, we can easily configure our operating system to play different sets of sounds for our actions — clicking a button can trigger anything from a snake's rattle to a lion's roar, without requiring the employ of a zoologist. But our keyboard and mouse always feel the same, no matter what we're doing.

This shortfall can partly be explained by the difficulty of simulating a haptic environment, in comparison to a visual or auditory one. This difficulty is due to several factors. Whereas images or sounds are perceived by humans through small apertures in the head, the haptic sense is highly distributed throughout the body. Even basic actions like grasping or exploring a surface make use of large numbers of mechanoreceptors in the skin, muscles, tendons, and joints of the hand and arm to create a vivid haptic experience. Completely addressing such a system is a complex undertaking. Secondly, whereas images and sounds consist of waves, tactile and kinesthetic stimuli take the form of physical forces. Newton's third law implies that in order to exert such forces, the exerting agent must be grounded with respect to the subject. This poses problems for device design and portability. Thirdly, we have a better scientific understanding of the visual and auditory sensory systems than we do of the haptic one — researchers are still discovering its basic properties.

Despite these obstacles, significant progress has been made in the field of haptic interfaces over about the last 15 years — innovative haptic devices have been produced, and compelling applications have been developed for them. Those applications can be roughly classified into two categories. The first is mainly concerned with the simulation of interaction with virtual worlds. Some of its notable members are surgery simulation, virtual sculpting, and the paradoxically named haptic visualization. This category has seen the most research activity and produced the most success.

While this first category is usually directly concerned with the perception of haptic stimuli themselves, the second is more interested in the potential that such stimuli hold for improving the efficiency of *interaction techniques*, or methods for entering information into a computer. While the potential for haptic feedback to support better interaction techniques is obvious, the haptic interaction techniques that have been developed to date exhibit nowhere near the rich haptic experience we have with our real world tools, despite significant research effort.

It must be noted that this gap is largely due to limitations in devices. Specifically, most available haptic devices support only *point force* haptic interaction, in which only the forces experienced by a single point in the virtual environment are displayed. On the other hand, our deftness with real world tools is thanks to the many points of contact and the variety of textures that we can rapidly and vividly perceive through the wide array of mechanoreceptors described above. Still, I believe that current point force technology is sufficient to support interesting and effective haptic interaction techniques.

One particular advantage of point force haptic simulation is that it allows the rendering of interactions not normally possible in the real world. For example, one can feel the inside of a solid object or experience a gravitational pull to an arbitrary point in space. While the design space is restricted by the physical limitations of the device, it is also expanded by such possibilities, provided that interaction techniques are designed with them in mind.

To appreciate the importance of such careful design, it serves to consider a typical factory at the dawn of the industrial revolution. Before electricity, a single steam engine was typically a factory's only non-human source of energy, and was thus located at the building's center, with a tangle of drive belts carrying energy from it to machines throughout the factory. This albeit clever design resulted in significant energy loss and layout constraints. When the more compact and efficient electric motor was invented, the need for a centralized energy source was negated. Nonetheless, this potential was not realized for some time, and many factories were built simply with a large electric motor at their centre in place of a steam engine [40].

I contend that the field of haptic interaction technique design is in a similar state — the full potential of the existing technology has yet to be realized. This thesis presents my initial efforts to help reduce that disparity.

In contrast to most earlier work, this work does not focus on haptic augmentations of existing graphical interfaces. Instead, haptic renderings are designed from scratch, using more abstract force-to-action mappings. Additionally, haptic feedback is displayed to the non-dominant hand, freeing the dominant hand to interact normally with the GUI using a mouse or other basic pointing device. Unfamiliar tasks or tasks requiring precise pointing are carried out by the dominant hand, and monitored using vision, while haptic feedback is used to reduce the attentional load and disruption caused by simple, repetitive tasks.

The guiding metaphor for this work is that of a motorist driving a car. The road ahead, which is unconstrained and more likely to present unfamiliar stimuli, is monitored visually. Meanwhile, for an experienced driver, most of the movements required to drive the car are controlled using haptic feedback: the steering wheel transmits the feeling of the road to the hands, and the forces applied to the brake and accelerator are monitored by the feet. In cars equipped with a manual transmission, the gearshift also provides important haptic feedback via the familiar gear channel constraint pattern. Simple, repetitive tasks like shifting the gears are controlled haptically, while complex, unfamiliar tasks like avoiding an obstacle are controlled visually. This division of tasks between the visual and haptic modalities has proven extremely effective, as it allows a complex and crucial task to be performed with a low attentional load.

In this thesis, the ideas represented by the above metaphor, along with other concepts gathered from both personal experience and the haptic literature, are assembled into a list of *design principles*. It consists of two parts: first, a set of properties of the human haptic and motor systems which are relevant to the area; and second, a set of specific guidelines for improving the effectiveness of haptic interaction techniques. While this list is tailored to my own research, I hope that elements of it may also appeal to other researchers in the field.

The other main contribution of this thesis is the design and evaluation of a haptic interaction technique, called *Pokespace*. The computing task for which it is intended is exemplified by applications featuring the familiar *tool palette* interaction technique, such as Adobe Photoshop and Microsoft Visio. In those applications, users modify a document by selecting a tool, adjusting its parameters, and applying it to a particular location. Despite both the relative simplicity and the high frequency of that task, executing it using a tool palette still requires several precise pointing movements and shifts of visual attention. I assert that this is analogous to having to look at the pedals of a car every time one wants to brake or accelerate — it is distracting, frustrating, and unnecessary. The goal of Pokespace is to eliminate the need for such distracting shifts of attention, making use of the application easier and more fluent.

This thesis is organized as follows: chapter two summarizes relevant work from fields related to the topic; chapter three lists the design principles mentioned above and introduces the Pokespace interaction technique; chapter four presents a user study investigating some basic effects of haptic feedback on bimanual interaction; chapter five describes two further user studies which evaluate Pokespace directly; and chapter six discusses conclusions and future research directions.

Chapter 2

Related Work

This chapter summarizes work related to this thesis in the fields of psychophysics, perceptual psychology, motor control, and human-computer interaction. First, the basic psychophysical properties of the human haptic system are introduced, and some of the relationships between the haptic and motor control systems are described. Next, a general taxonomy of available haptic simulation technology is given, followed by a summary of the notable applications that have been developed for it. Finally, the dichotomous concepts of the *interaction task* and *interaction technique* are defined, and research in the field of haptic interaction technique design, the focus of this thesis, is examined.

2.1 The Human Haptic and Motor Systems

The field of haptic human-computer interaction implies a necessary duality of interest in both the human haptic and motor systems — the former by definition, and the latter since any interaction involves some movement. Not surprisingly, the two systems are substantially co-dependant. For example, the haptic system depends on the motor system to control the active exploration of objects, while the motor system depends on the haptic system to report the locations of the limbs in space as a movement is being executed. As such, knowledge of both systems is crucial to the design of haptic interaction techniques. This section describes several of their relevant capabilities and associated phenomena.

The neurophysiology of the haptic system is based around the body's *mechanoreceptors*, which are the components of the nervous system believed to be responsible for transducing mechanical energy into neural signals [30]. Several types of mechanoreceptors have been identified, and their locations and roles suggest a logical division of the sense of touch into two separate perceptual systems. The *cutaneous* or *tactile* system relies mainly on the four different types of receptors located near the surface of the skin, and is responsible for perceiving the sensations of pressure, position, spatial acuity, temperature, and pain [30]. Meanwhile, the *kinesthetic* system is composed of receptors located in the muscles, tendons, and joints. That system is chiefly concerned with sensing the position of the limbs in space, the exertion of forces by the muscles, and the external forces on the limbs. It should be noted that the correspondence of mechanoreceptors to haptic subsystems is not absolute; for example, sensations of skin stretch can contribute to the kinesthetic sense of position [10].

Sensations from both systems combine to form a unified haptic percept. For example, in lifting an object with our hand, we experience pressure and texture via our tactile sense, and weight via our kinesthetic sense. Both are indispensable. Seminal work by Lederman and Klatzky [37], [38], identified eight "exploratory procedures" typically used to explore and identify objects. Many of them clearly make use of both tactile and kinesthetic information.

Also related to the exploration and identification of objects is work by Kirkpatrick and Douglas [29]. In that study, participants were asked to identify an object as one of five simple shapes using a simulation device which allowed only one point of contact with the object. Participants took an average of 22 s to perform this otherwise simple task. This work agreed with earlier findings of Klatzky, Loomis, Lederman, Wake, and Fujita [31], in which participants were progressively slower to identify real shapes with their hands as the number of points of contact was decreased. These findings indicate that humans are poor at integrating spatial information perceived over time into a valid whole, and that the many points of contact which operate in parallel are crucial to our ability to identify objects.

In situations where the objects we wish to explore are not immediately available to our skin, we sometimes exploit the phenomenon of *distal attribution* [41]. That term refers our ability to effectively extend our sense of touch through an object we are holding, so that attention is paid not to the feeling of the object, but to its contact with the rest of the environment. Examples of this phenomenon are widespread, from the feeling of a sheet of paper through a pen, to the feeling of the road through an automobile steering wheel. This ability is of obvious importance to the design of some haptic simulation devices.

Just as tactile and kinesthetic information is often combined to form a single haptic percept, so too is information from the visual and haptic modalities. The study of such visual/haptic interactions has revealed some interesting properties of the relationship between the two senses. For instance, early investigations revealed that when the view of an object was distorted so that it looked smaller than it felt, the object was perceived as small. These results gave rise to the term *visual dominance.* But later work found evidence that the contributions from each modality differed depending on the nature of the task, and could therefore be modulated by attention [30].

Several results in the perceptual psychology and motor control literatures indicate that the haptic and motor systems are able to share information quite rapidly in certain situations. A direct comparison of the reaction times to simple stimuli presented to the visual and haptic modalities revealed an interesting result. Reaction times to a tactile stimulus presented to the finger are lower than to a visual stimulus indicating the same finger, as long as the finger to be moved was the same finger that was stimulated [7]. Another striking example is our ability to grip delicate objects without breaking them [27].

2.2 Haptic Devices

The complexity of the human haptic system, so sensitive and widely distributed throughout the body, presents a formidable challenge for anyone wishing to fool it. This unique challenge has inspired a considerable body of work in the development of haptic display and simulation technology, resulting in the creation of a wide variety of devices.

But unlike visual and auditory display technology, the stimulus production capabilities of haptic devices do not yet match the sensing capabilities of the human haptic system. The field of haptic human-computer interaction is therefore considerably constrained by the hardware itself. In order to make most effective use of that hardware, an understanding of the properties, strengths, and weaknesses of the range of existing devices is necessary. I present an overview in this section.

Haptic devices can be classified according to the type of stimulus they present: tactile or kinesthetic.

2.2.1 Tactile devices

Tactile stimuli are those perceived by the surface of the skin. Therefore, tactile devices usually interact with the fingers or hand, where tactile sensitivity is the highest. They also typically involve very small forces, designed to perturb the skin, not to move the limbs.

Shimoga [59] identified five main approaches to tactile simulation: visual display, neuromuscular stimulation, pneumatic stimulation, vibrotactile stimulation, and electrotactile stimulation. Visual display, somewhat contradictorily, involves the suggestion of a tactile experience via the visual sense. This approach is neither common nor effective. At the other extreme, the neuromuscular approach involves the direct issuance of neurological signals to the neuromuscular system. Such an approach has the potential to be very effective, but involves such high liability and invasiveness that it has not received much attention. Related to the neuromuscular approach is the use of electrotactile stimulation, whereby small electrical currents are applied to the skin, evoking a tingling sensation at the site of the stimulation.

The pneumatic and vibrotactile approaches are the only two which involve the application of real forces to the skin. Pneumatic forces are delivered either by directing jets of air at the skin or by varying the pressure of inflatable bladders placed next to it. Vibrotactile forces are delivered through physical contact between the skin and a vibrating agent.

Of Shimoga's five categories, vibrotactile devices are by far the most common, since they are relatively easy to construct and present the least encumbrance to the user, while still providing fairly high stimulus quality. Perhaps the most common vibrotactile device is the vibrating ringer built into most modern mobile telephones. Such a device usually consists of a single weight oscillated by an electric motor, which vibrates the entire unit. Most video game console controllers now also contain similar devices. More sophisticated devices consisting of multiple vibrating units over the surface of the fingers, hand and elsewhere on the body have also been developed [63].

The above five categories do not account for devices which convey stimulation through physical contact with a non-vibrating physical agent. For example, so-called *shape devices* simulate physical shapes using a dense array of small pins. Each such pin is individually controllable, and moves either orthogonally to the skin [67], or laterally [22]. Braille simulation devices also fall into this category.

A further notable example of this sixth category is the device constructed by Akamatsu and Sate [2] consisting of a regular mouse augmented with a single electromagnetically actuated pin which protruded through the left button of the mouse. That device was used to provide tactile feedback related to the pointing movements of the user.

2.2.2 Kinesthetic devices

In contrast to tactile devices, kinesthetic devices are those which interact primarily with the kinesthetic receptors found in the muscles, tendons, and joints. Such devices thus generally involve larger forces than their tactile counterparts, and are more concerned with moving or resisting the movements of the limbs rather than deforming the skin.

Like tactile devices, kinesthetic devices can be classified according to several criteria. One such criterion concerns the number of degrees of freedom along which the device can exert forces. A popular example of a one degree of freedom (1-DOF) device is the haptically actuated rotary knob [65]. Examples of 2-DOF devices are numerous, and include commercial products like the Logitech Wingman force feedback mouse and joystick, as well as research prototypes such as the Pantograph [54] and the Moose [50].

More sophisticated 3-DOF and 6-DOF devices are also popular. The most popular such device is the SensAble Phantom, of which both 3-DOF and 6-DOF models are sold. A 3-DOF version was the device used for the work in this thesis.

The end-effector of the Phantom is a pen-like *stylus*. Within the mechanical limits of the structure, the stylus can be moved freely in three dimensions with relatively little inertia. Forces are applied to the stylus with three electric motors attached by cables to the device's structure. In turn, the position of the stylus tip is sensed by three optical sensors. The Phantom model used in this thesis is described as a wrist device, in that its working volume is sufficient to allow a full range of motion pivoting at the wrist.

The 3-DOF Phantom is a *point force* device. That term, coined by Massie [44], refers to the fact that the device is limited to the simulation of the forces experienced by a single infinitesimal point in a virtual environment. In the real world, this scenario is physically impossible, since any point of contact will be subject to some friction resulting in a torque. However, the metaphor is convenient since it allows for a much simpler design and is capable of quite convincing effects.

Devices can have many more degrees of freedom, such as the 11-DOF Immersion CyberForce whole-hand-and-arm device, which has one DOF for each finger in addition to the six for the translational and rotational dimensions of force feedback applied to the whole hand.

As discussed above, kinesthetic devices exert substantial forces, and therefore must somehow be grounded. This grounding is variously achieved in two different ways. The more straightforward and common design is that in which the end-effector is either physically fastened to the earth, or anchored by a sufficiently heavy base. The latter is the case for the Phantom device, as shown by the large black box below the device's armature. All the devices listed in the previous paragraph fall into this category. Such devices are easier to engineer, but sacrifice portability, since the user is limited to the workspace dictated by the limits of the device.

The alternative to this design is to ground the end-effector to another site on the user's body. Such is the case with the Immersion CyberGrasp haptic glove, which attaches to the hand and provides one DOF of force feedback for each finger. The rods which exert forces upon the fingers are connected to a plate on the back of the hand, thus providing the necessary grounding. This design overcomes the portability issues of earth-based devices, but is far more complex to engineer, given the irregularity of human body. Furthermore, the grounding site on the body is unavoidably subject to a force opposite that applied to the end-effector, potentially compromising the effectiveness of the simulation.

Another distinguishing characteristic of the kinesthetic device is the method of control. There are two major such methods: *impedance control* and *admittance control.* The more popular method is impedance control, in which the user moves the device, and the device responds with a force if appropriate. In terms of the control loop, this corresponds to "position in, force out". Admittance control is the opposite — the user exerts a force on the device, and the device reacts by moving to the appropriate position. Naturally, this duality carries certain tradeoffs. For example, the simulation of open air is automatic for impedance control, whereas an admittance controlled device must accelerate very quickly at the lightest touch to provide the illusion of free space. The opposite tradeoff results for the simulation of stiff surfaces, which is easier to achieve with admittance control, whereas surfaces feel unavoidably *spongy* with impedance control.

Also related to the perception of stiff surfaces is the notion of a device's refresh rate. Just as a television screen must show about 30 frames per second to provide the illusion of a continuous picture, most kinesthetic haptic devices must operate at a refresh rate of about 1000 Hz to ensure convincing haptic effects. Lower refresh rates are particularly detrimental to the simulation of stiff surfaces for impedance control devices. This requirement implies that all computations required to drive a haptic simulation must be performed in less than one millisecond, which can present a formidable challenge for algorithm design, even with modern computing systems.

An interesting type of haptic device is that designed solely to be used as an experimental apparatus. A notable example is the kinesthetic device built by Robles-de-la-Torre and Hayward [56] in their landmark study of shape perception through active touch. The device was precisely crafted to simulate the exploration of a shape with the index finger. The study revealed that properly designed lateral force cues can fool a user, for example, into perceiving a convex object as concave, and vice-versa.

2.3 Haptic Human-Computer Interaction

The steady progression in the quality of haptic devices has created along with it a growing interest in the development of applications for that technology. That field is referred to here as haptic human computer interaction (HCI). One of the earliest forms of haptic HCI was employed in the design of aircraft controls. In early aircraft control systems, the pilot's controls were mechanically linked to the control surfaces on the aircraft's wings. This direct link carried valuable haptic feedback to the pilot. For instance, as the aircraft approached a stall, an aerodynamic buffeting resulted and was haptically perceivable. When larger aircraft were developed, mechanical linkages gave way to electronic servo systems, thus eliminating the direct haptic feedback to the pilot. To alleviate this problem, the pilot's controls were subsequently outfitted with "stick shakers", made to vibrate when the aircraft was in danger of stalling.

This innovation was an example of a now common area of haptic interaction called environmental monitoring, which exploits the ecological monitoring role of the haptic sense. Environmental monitoring applications are characterized by the need to alert the user of some event when attention is directed elsewhere. The surface of the skin is particularly well suited to such monitoring when vision and audition are either not available or inappropriate. Other examples of haptic environmental monitoring range from vibrating mobile phone ringers to military applications.

Considerable efforts have been made to bridge the gap between basic psychophysics and haptic HCI through the in-depth study of psychophysical phenomena of particular interest to haptic HCI. A notable example of this basic research is the work of MacLean and Enriquez on haptic icons [43]. That work proposed to build an abstract expressive haptic language based on short vibrotactile pulses of varying amplitude, frequency, and wave shape. Several experiments sought to maximize the discriminability and individual salience of the stimuli. Other work by Swindells, MacLean, Booth, and Meitner [64] explored users' affective responses to simple haptic environments.

Just as rendering for graphical display has been a core area of graphics research, so too has general purpose rendering been an active area of haptic HCI, especially for point force devices. General purpose rendering is concerned with mathematical techniques for maintaining an internal computing model of a virtual environment, and computing the appropriate forces which should be displayed to the user based on their interaction with that environment. A canonical technique for haptic rendering of rigid polyhedra using a "proxy object" was developed by Zilles and Salisbury [68]. That model dealt only with point force 3-DOF interaction. Later work, such as that of Otaduy and Lin [51], incorporated 6-DOF interaction, though the computational cost of the best such algorithms is still quite high.

Another early instance of haptic HCI, called telemanipulation, involves the remote control of a robotic agent, called a *slave*, using a local device, called a *master*. Typically, the slave is used to perform tasks in inhospitable environments, such as deep water exploration or bomb disposal. Haptic feedback is used to communicate the forces encountered by the slave to the master, with the goal of making control of the slave more natural and fluent. Some of the first such work was conducted in the 1950s at Argonne Nation Laboratories by Goertz and Thompson [16].

The simulation of interaction with virtual environments has been an extremely productive area of haptic HCI, and many of the most successful commercial applications of haptics research are instances of it. Perhaps the most successful of those has been the field of medical training, in particular surgical simulation. Such applications attempt to provide realistic surgical training to doctors without any risk of trauma to a real patient. Their success is no doubt due in part to the compatibility of the task with existing stylus-based haptic technology. A recent article by Delingette and Ayache [11] provides a survey of this area.

Another commercially successful virtual environment application is virtual sculpting. The task of sculpting is a highly haptic one, since it involves fine motor movements, an intimate knowledge of material properties, and frequent occlusion by the hands or the sculpture itself. The FreeForm sculpting application developed by SensAble Technologies allows virtual sculpting with haptic feedback via the Phantom haptic device. It has been used for product design by companies such as Adidas and Hasbro.

One of the earliest virtual environment applications was project GROPE, conducted by

Brooks, Ouh-Young, Batter, and Kilpatrick [8], which used haptic feedback to display molecular interaction forces in a simulated molecular docking task. Project GROPE represents early work in the paradoxically named field of *haptic visualization*, where haptic feedback is used to provide insight into data which may be difficult to interpret visually. Avila and Sobierajski [3] later developed an often-cited method for haptic volume visualization, which allows solid objects to be explored internally using point force haptic feedback.

Most of the above research is devoted to specialized tasks which are difficult or impossible without haptic feedback. But the importance of haptic feedback to our interaction with ordinary objects and tools in everyday life suggests that haptic technology also has the potential to support more fluent and efficient performance of routine computing tasks, more formally referred to as *interaction tasks.* The development of haptic *interaction techniques* for those tasks has motivated a considerable amount of research in haptic HCI, including this thesis.

2.4 Interaction Tasks and Techniques

The most widely recognized definitions of the terms *interaction task* and *interaction technique* were offered by Foley, vanDam, Feiner, and Hughes [14, p. 349]. The former was defined as "the entry of a unit of information by the user", and is further divided into four *basic interaction tasks*, namely positioning, selecting, entering text, and entering numeric quantities. Thus defined, an interaction task is independent from the manner in which it is accomplished. An *interaction technique*, on the other hand, was defined as the method used to perform an interaction task, including both hardware and software.

Since new interaction tasks seldom come about, most HCI research in the area is devoted to the development of novel interaction techniques, using both new hardware devices and new software constructions. The history of such developments is dotted with several landmarks which deserve mention here.

Though we now often laugh about it, interaction with early computers was rather tedious. Input was accomplished with punch cards, and output with line printers. As Norman [46] would put it, the gulfs of execution and evaluation were enormous. Later came CRTs, keyboards, and command line interpreters, as the gulf began to shrink. The introduction of the mouse for input and bitmapped screen for output produced a highly versatile combination which vastly expanded the user interaction design space. It was only then that interest in interaction techniques began to blossom. This led to the invention in the late 1970s of the graphical user interface (GUI) which, as opposed to a blank command prompt, provided the user with graphical depictions of objects, such as windows, icons, menus, and pointers, around which real world metaphors of interaction could be constructed. Early systems featuring a GUI, notably the Xerox Alto and Star, and the Apple Lisa, were some of the first systems to support *direct manipulation.* That term was coined in 1983 by Shneiderman [60] to describe a then emerging style of interface which supported "visibility of the object of interest; rapid, reversible, incremental actions; and ... direct manipulation of the object of interest."

Since then, the research of interaction techniques has continued to grow as more and more innovative input and output devices are produced, and the computational capabilities of existing systems increases. Some of the most intriguing interaction techniques resulting from this work are those which have moved beyond real world metaphors to take full advantage of the flexibility of the virtual worlds which computers can portray. *Radial menus* and *Toolglass* are two celebrated such techniques.

Radial menus are like the familiar linear context menus found in most modern operating systems, with one crucial difference: the menu choices are displayed in a pie-shaped arrangement, with one wedge of the pie for each available command. Several incarnations of the radial menu concept have been proposed, including pie menus [25], marking menus [34], control menus [53], and FlowMenus [20]. In most designs, a command is selected by moving the pointer beyond the boundary of the wedge of the desired command. Some selections may trigger the appearance of a sub-menu, as with linear menus. Thus, a complete selection takes the form of a gesture or *mark* along a path defined by the chosen wedges. Studies have shown that such marks can be learned and executed without reference to the graphical menu display [35].

This design gives rise to the principle of *rehearsal*, first coined by Kurtenbach [36]. For interaction techniques designed to emphasize rehearsal, the novice use involves the same gesture as the expert use. This makes the transition from novice to expert smoother, since the novice is naturally training for the expert use simply by using the system. This approach improves on previous techniques such as the shortcut or accelerator key, since learning an accelerator key mapping requires conscious self-training, which many users avoid. Moreover, users who were once experts who return to using a system after a long period of inactivity also benefit from rehearsal, since they can more easily reacquaint themselves with the interface.

The Toolglass interaction technique consists of a translucent *sheet* of *widgets*, each of which represents a certain command [6]. The sheet is manipulated using the user's non-dominant hand, while a standard mouse pointer is controlled with the dominant hand. A command is applied to a given location or object by positioning the appropriate widget over the location and *clicking through* it. A study by Kabbash, Buxton, and Sellen [28] reported that Toolglass performed significantly better than the traditional tool palette for a simple connect-the-dots line drawing task.

This approach carries several benefits, one of which is the elimination of *temporal modes.* A temporal mode is a particular state of an interface which is activated by a user action and persists until deactivated with a separate action. The familiar tool palette is a common example — a tool is selected with a mouse click, and remains active until another mouse click selects a different tool. Raskin [55] identified a crucial problem with temporal modes, in that humans are notoriously prone to forget which mode they have activated, resulting in *mode errors.* As such Raskin advocated the avoidance of modes altogether. Effectively, Toolglass replaces temporal modes with *spatial modes*, and the fallibility of human memory is removed from the equation.

Radial menus and Toolglass are both examples of interaction techniques that allow users to both select a command and proceed with direct manipulation of the object of interest in a single motion. In a recent study, Guimbretière, Martin, and Winograd [19] compared three interaction techniques featuring that characteristic: Toolglass, control menus, and Flow-Menus. They found that Toolglass was significantly slower than both menu techniques, and thus concluded that the key factor in its previous successes was its unification of command selection and direct manipulation, not its two-handedness.

The results of that study were surprising given the enthusiasm and encouraging results surrounding two-handed or *bimanual* interaction in the HCI literature up to that point. For example, foundational work by Buxton and Myers [9] suggested that users were capable of simultaneously providing continuous input from both hands without significant overhead. Later work involving a bimanual neurosurgical imaging interface [23] found that performance was best when the dominant hand operated relative to the frame of reference of the non-dominant hand in an asymmetric task. These results agreed with the often cited *kinematic chain* theory of Guiard [18]. Still later work by Leganchuk, Zhai, and Buxton found superior performance for a bimanual technique in an area sweeping task [39]. They suggested that bimanual interaction had the potential for both manual benefits, due to increased time-motion efficiency, as well as cognitive benefits, due to the reduction in the mental workload required to compose and visualize tasks. Taken together, these studies suggest that bimanual interaction is a worthwhile enterprise, but that its utility is dependent upon both the particular design of the technique and the task for which it is employed.

A central theory to interaction technique design is Fitts' Law, which predicts the time required to make basic pointing or reaching movements. It states that the average time required to move to or *acquire* a target of width W and at distance D is

$$a + b \log_2 \frac{2D}{W}$$

where a and b are scalar terms. The main term of the formula, $\log_2 \frac{2D}{W}$, is referred to as the *Fitts index of difficulty*, as movement time scales linearly with it [58]. While Fitts' original experiments involved direct pointing movements with a handheld stylus, the results have also proved valid for indirect pointing movements, such as those performed with a computer mouse [42].

A second, more recent theory concerns the relationship between the perceptual structure of an interaction task and the control structure of an input device. That work, by Jacob, Sibert, McFarlane, and Mullen [26], extended an earlier theory [15] from the perceptual psychology literature which held that the dominant perceptual structures of the attributes of objects in multidimensional spaces can be placed along a continuum defined by two extremes: attributes which combine perceptually to form a unified whole are considered *integral*, while those which remain perceptually separate are *separable*. For example, the hue and saturation of the color of a shape are integral, but the color and size of the same shape are separable. Jacob et al. applied this idea to HCI by suggesting that the control space of an input device should match the structure of the perceptual space of an interaction task to which it is applied. For example, the theory predicts that a pair of 1-DOF sliders would be poorly suited to specifying a two-dimensional position, but well suited to inputting the size and color of an object.

2.5 Haptic Interaction Techniques

The advent of haptic interface technology suggested a next logical step in the field of interaction technique design, and the design and evaluation of haptic interaction techniques is now a recognized sub-field of both the haptic and HCI communities. But note that the dividing line between the field of haptic interaction techniques and other areas in the haptics literature is blurred. Broadly speaking, the haptic rendering of virtual environments could be considered an interaction technique for the task of virtual environment simulation. However, for the purposes of this thesis, a haptic interaction technique is considered to be a system in which the primary role of the haptic feedback is to enable more fluent or efficient interaction, rather than to provide a consciously perceived haptic sensation.

The first efforts at incorporating haptic feedback into the user interface involved elementary pointing and steering tasks. Akamatsu, MacKenzie, and Hasbroucq [1] used a mouse that provided tactile feedback when users had entered a target. They found that users required less time to verify that they had reached the target, but that overall pointing time was not significantly improved. Engel, Goossens, and Haakma [13] reported similar findings with a trackball modified with two servo motors, one for each axis. Dennerlein, Martin, and Hasser [12] found that haptic feedback improved performance times by 52% for a steering task in which the cursor was moved along a two-dimensional 'tunnel' to a target.

Later, Oakley, McGee, Brewster, and Gray [49] used the more sophisticated Phantom haptic device to apply four haptic effects to standard GUI buttons: *texture*, which consisted of a radial sinusoidal pattern, similar to ripples on a pond; *recess*, in which a haptic depression was rendered at the location of the button; *friction*, which used a Coulomb-like friction model to simulate a frictional surface at the location of the button; and *gravity well*, which rendered a 0.5 N force attracting the user to the center of the button once they moved over it. They found that the haptic enhancements reduced errors, but pointing time was not significantly improved.

The concept of *virtual fixtures* was introduced by Rosenberg [57] as a method to improve performance in teleoperation. In his system, planar constraints were *overlaid* on top of the force feedback reflected from the remote environment. The constraints were arranged in various orientations with the goal of improving performance in a simple peg-in-hole pointing task. He found that haptic fixtures improved performance by as much as 59%. The notion of virtual fixtures has since been employed by several authors [52], [5].

Since all the above research dealt with pointing and steering tasks with only one potential target, its applicability is questionable. After all, if there is only one target, there is no need to point at it — the action can be taken automatically by the computer. Oakley, Adams, Brewster, and Gray [47] argued that in a more realistic interface with multiple targets, forces associated with unintended targets can disturb the movement of the user, reducing performance. Oakley, Brewster, and Gray [48] addressed this concern in their experiments with haptic augmentations of a menu technique. Instead of calculating the applied force based only on the user's position, they modulated the magnitude of the haptic effects according to the speed of the user's movements. They reported a significant reduction in error rates with no significant difference in speed, compared to a visual-only technique.

In another example of selecting from multiple targets, Komerska and Ware [33] added haptic effects to two-dimensional menus in a three-dimensional virtual reality environment. The effects included constraining the cursor to the plane containing the 2D menu, constraining the cursor to within the menu's boundaries, and snapping the cursor to the centre of a menu item. While other aspects of their technique were more successful, their haptic enhancements only improved performance 0–4%.

Miller and Zeleznik [45] added haptic effects to GUI features, such as window borders, buttons, and checkboxes. In a similar vein, Komerska and Ware [32] extended the 3D interaction techniques of their GeoZUI3D system. In both projects, forces were added to either attract the pointer towards a target or keep the pointer on a target once acquired. Neither of these papers reports empirical studies of its designs.

Two groups have developed bimanual haptic interaction techniques. Using a device they developed, Bernstein, Lawrence, and Pao [4] designed a technique featuring haptic feedback to the dominant hand in the form of contact cues and snap-to-grid effects for a 3D object editor. No evaluation of these effects was reported. Grosjean, Burkhardt, Coquillart, and Richard [17] added vibrotactile feedback to a technique for selecting one of 27 commands in a virtual reality environment. Each command was mapped to a region in a space divided in the manner of a Rubik's cube, and users felt a vibration every time they crossed a threshold from one region to another. Unfortunately, the tactile feedback produced slower performance than the same technique with none.

Although creative and well-executed, only one of the above haptic techniques was found to support lower completion times. Several others reduced errors, a result of practical value, but not an indication of substantially higher fluency. Arguably, this absence of compelling results has thrown the utility of haptic feedback to interaction technique design into question. However, it should be noted that nearly all the above techniques were simply haptic decorations of existing interface components. As the factory analogy of the previous chapter suggests, such an approach does not allow a complete exploration of the design space.

One notable departure from this trend is the work of Snibbe et al. [62] in designing haptic interaction techniques for the manipulation of digital media such as film or voicemail. They constructed their systems "as physical task metaphors, rather than as literal representations of the media". For example, their *Alphabet Browser* technique used haptic and auditory feedback to explore a list of music tracks. Rotating a haptically actuated knob scanned through the tracks. Each time a track was passed, a haptic detent was felt, and the artist's name was spoken. This technique is not based on any existing graphical user interface design, but instead uses a metaphor relating the haptic feedback directly to the content being manipulated.

That approach is most similar to the one taken in this thesis, which contends that haptic interaction techniques should be designed from scratch, and in explicit consideration of the properties of both the haptic device and the human haptic and motor systems. If used in this manner, I believe there exists tremendous potential for haptic feedback to enable more efficient and enjoyable interaction. The remainder of this thesis describes my work in that direction.

Chapter 3

Design Principles and Implementation

The importance of the haptic sense to our remarkable manual dexterity has already been discussed at length in this thesis, as has the conspicuous lack of sophisticated haptic feedback in the modern human-computer interface. Granted, to say that the typical computing experience is totally devoid of haptic feedback is to overstate. The keyboard affords a particularly rich haptic experience: its buttons are curved to match the human finger pad, have sharp edges and valleys between them, have special bumps on the two primary home keys, and make a soft click when pressed. Thus, in the context of the taxonomy of Foley et al. [14], the interaction task of text entry is already quite haptic. But the other three of their interaction tasks — positioning, selecting, and entering numeric quantities — are most often performed with the mouse, which offers a much simpler haptic experience.

The above should not be read as an indictment of the mouse. In fact, the mouse is well suited for what I refer to as *general purpose pointing*, or the ability to point with near-single pixel accuracy, which the modern WIMP-style user interface demands. As discussed in chapter 2, using haptic feedback to improve general purpose pointing does not seem fruitful, and the standard mouse remains the most effective method to date. Therefore, the approach followed in this thesis was to leave the role of the mouse unchanged in the dominant hand, and to display haptic feedback via the non-dominant hand instead.

The device used to do so was the SensAble Phantom. While the Phantom is a highly

---

Portions previously published [61]. Copyright 2006 ACM Press. Adapted with permission.

sophisticated and widely available machine, it comes with its own set of limitations, as discussed in chapter 2, which restrict the design space of interaction techniques employing it. On the other hand, the use of the non-dominant hand expands the design space considerably. With the demands of general purpose pointing eliminated, the entire haptic working volume is freed for haptic effects. Also, since there is no longer a need to present a correspondence to the two-dimensional visual display, the three-dimensional capabilities of the Phantom become fully exploitable.

In consideration of all the above, I feel that there remains significant potential for effective interaction techniques based on point force haptic display. This chapter presents my initial efforts to realize that potential.

3.1 Design Principles

The work for this thesis began with a simple idea: that haptic feedback could make everyday interaction with computers more efficient and enjoyable. Subsequently, over the course of a review of the literature, the design of several prototypes, and some informal experimentation, I assembled a set of design principles, which are presented here in two categories. First, several especially relevant *properties* of the human haptic and motor systems are listed, followed by a set of *guidelines* for haptic interaction technique design.

Note that these principles are simply suggestions, and do not represent empirically verified propositions. It is expected that they will continue to be refined as further development and user experimentation are conducted.

3.1.1 Haptic/Motor Properties

1. *Human temporal integration skills are poor.* Temporal integration here refers to the ability to combine spatial information received serially over time into a single picture. Work by Kirkpatrick and Douglas [29] and Klatzky et al. [31] implies that this ability is not very prominent, at least for the haptic modality. In both studies, it was found that subjects had difficulty identifying simple shapes when they had few points of contact with which to explore them. Instead of the parallel processing of many contact points, subjects presumably had to rely on integration of signals from few contact points over time from working memory, and performance dropped.

2. *Movement of the limbs is integral.* This property invokes the ideas of integrality and

separability described by Jacob et al. [26]. Their definition was with respect to input devices, and they defined an integral device as one in which "movement is in Euclidean space and cuts across all the dimensions of control". Here, we apply the same notion of integrality to human limb movements — limbs do not naturally or easily move separately along the axes of the Euclidean coordinate system.

3. *Human motor memory skills are good.* Motor memory is a striking phenomenon, best exemplified by some of the most common movements in everyday life. The act of getting into a car is an especially powerful example. When one first buys a new car, the process of opening the door, getting in, getting comfortable, buckling the seatbelt, turning the ignition, engaging the gears, and driving away, can be tedious and time consuming. But with sufficient practice, all those tasks are soon performed in a single, fluid action, taking only seconds, and with very little mental workload.

3.1.2 Design Guidelines

1. *Design for rehearsal.* Interfaces support rehearsal when their novice use involves the same movements as their expert use, instead of requiring users to consciously choose a new strategy to improve performance. This exploits motor memory and makes the transition from novice to expert smoother. Haptic feedback offers a unique way to support rehearsal by allowing control of interaction to transfer from the visual modality to the haptic one, just as a motorist first has to look to buckle their seatbelt, but eventually learns to do so by feel. The visual modality is then free for other, more complex tasks.

2. *Design for gestures.* The power of human motor memory is most evident in highly practiced actions, such as tieing one's shoes, or playing a quick sequence of notes on a piano. Such actions, sometimes referred to as *gestures*, are performed very rapidly, and with little cognitive effort. They also often involve rich haptic cues. Accordingly, haptic interaction techniques should aim to support the development of gestures.

3. *Keep the haptic environment simple and consistent.* Since vision is more precise, and better at conveying abstract symbols, it should be used to guide performance of more complex, finer, or more unfamiliar tasks. The haptic modality, on the other hand, is best suited to simpler, coarser, or better practiced tasks. The user should not be

expected to interpret or react to unfamiliar stimuli via the haptic modality. Instead, the haptic environment should be kept consistent and simple enough to support the development of motor memory.

4. *Maximize the number of backstops.* A *backstop* is a physical constraint rendered by the haptic device, such as a wall or a corner. The use of backstops has a two-fold advantage. First, a backstop can physically guide the user to a target, effectively reducing the precision required for the movement, just as the concave shape of an automobile ignition facilitates the insertion of the key. This allows the user to make faster movements, and compensates for poor temporal integration abilities. Second, the distinct and immediate haptic feedback provided by the backstop serves as confirmation that the movement has reached its target, thus shortening the overall movement time. Note that a backstop could be considered a type of virtual fixture [57].

5. *Promote visual attention on the object of interest.* Many computing applications involve both an object of interest, such as a canvas or a document, and a set of controls for manipulating the object. Since the object of interest is usually more complex and unfamiliar than the controls, visual attention on the object of interest should be promoted by using haptic feedback to support the manipulation of the controls, just as a motorist changes the gears and controls the pedals by feel while watching the road.

6. *Support integrality of movement.* Since human limb movements are integral with respect to Euclidean space, interfaces should not require the separation of movements along the dimensions of that space. Where the logic of the interface makes such movements appropriate, they can be enforced using haptic feedback.

7. *Make optimal use of the device's degrees-of-freedom.* The two-dimensional nature of the visual display should not restrict the haptic interaction technique design space. In particular, the third degree-of-freedom of the Phantom offers considerable possibility for a rich haptic environment. This potential should be exploited.

8. *Use the non-dominant hand for haptic feedback.* As stated above, the mouse is well suited to general purpose pointing, and haptically augmenting the mouse causes problems. The haptic device should be held with the non-dominant hand, and the mouse should remain in the dominant hand.

The rest of this chapter describes a common interaction task, and an interaction technique developed according to these guidelines.

3.2 The Interaction Task

I have argued at the start of this chapter that of the three general interaction tasks chiefly performed by the mouse in current interfaces — positioning, selecting, and entering numeric quantities — positioning seems to be the best suited to the mouse, especially where fine manipulations are necessary. On the other hand, selecting and entering numeric quantities do not generally require the same level of precision.

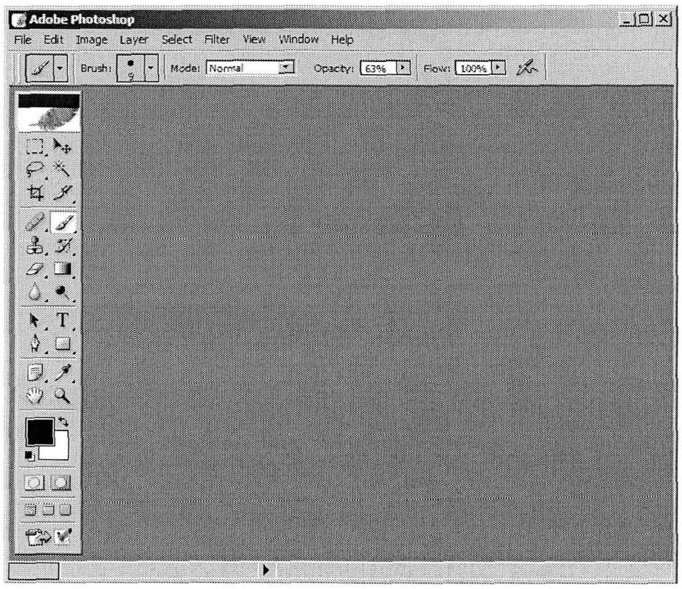
Those two tasks are the ones focused on in this thesis. More specifically, I am concerned with a composite task referred to here as *command selection and parameter specification.* This task is most commonly found in design applications such as Adobe Photoshop or Microsoft Visio, in which an object of interest (such as a drawing or diagram) is modified by a series of applications of different commands, usually referred to as *tools.*

Interaction with such applications usually follows a sequence: first a tool is selected (a selecting task), its parameters are adjusted (an entering text or entering numeric quantities task), and it is applied to the object of interest at a certain location (a positioning task). Note that this sequence is not always strictly followed — some tools have no parameters to adjust, and sometimes a tool will be applied several times before a different tool is selected. Nonetheless, it is the first two stages of that sequence which constitute the task of command selection and parameter specification.

The most common interaction technique addressing this task is the *tool palette*, in which tools are represented by icons and grouped together to form a *palette* located towards the side of the screen. The parameters for those tools are generally controlled by a set of standard GUI widgets also grouped together and located near the side of the screen. Both the tools and the parameter controls are operated using the mouse.

A typical tool palette interface is shown in Figure 3.1. The user has just selected the paintbrush tool, and is about to modify the brush size using the drop-down parameter control near the top of the screen.

While it is well proven, the tool palette technique has several disadvantages. It requires the user to shift their gaze away from the object of interest to the palette, and then shift a



Brush: 9
Mode: Normal
Opacity: 63%
Flow: 100%

Figure 3.1: A typical tool palette interface, consisting of a collection of tools on the left, along with a set of controls for manipulating the parameters of the selected tool along the top. (Adobe product screen shot reprinted with permission from Adobe Systems Incorporated).

second time to the dialog box. The palette and dialog box are often complex and tightly-packed, requiring considerable attention to interpret and care in aiming the pointer to select the correct entry. The palette and dialog may also be far from the object of interest. This combination of small targets and longer movement distance gives the movements a high Fitts index of difficulty. Also, the tool palette technique fails to leverage the repetitiveness of the task to support the development of motor memory. No matter how many times a user has selected a given tool, a visually controlled, relatively precise, closed-loop pointing movement is required to do so.

At the same time, the task of command selection and parameter specification seems well suited to haptic augmentation of the sort suggested by the principles in the previous section. As stated in the previous paragraph, it involves a lot of repetition, since the number of tools is limited, and the number of frequently used tools is even smaller. The opportunity for the development of motor memory thus seems high. Secondly, the task requires precise pointing for the application of tools to the object of interest, so the presence of the mouse is beneficial. Finally, the task often involves a significant amount of attention on the object of interest, and thus would likely benefit from a reduction in the distraction caused by the interface controls.

3.3 Pokespace

Pokespace is an interaction technique designed as an alternative to the tool palette technique described in the section 3.2, and inspired by the guidelines described in section 3.1.2. It consists of a mapping of a set of tools and their parameters to a three-dimensional dynamic force field, displayed and manipulated with the SensAble Phantom haptic device. The Phantom is held in the non-dominant hand, while the mouse remains in the dominant hand and specifies the location at which the tools are applied.

Implemented in this thesis are three common tools: brush, text, and zoom. The current prototype implementation does not include the actual functionality of those tools, only the ability to select them and modify their parameters.

3.3.1 Basic Haptic Rendering

Within the Phantom's workspace, each tool is mapped to a plane perpendicular to the z axis (parallel to the screen). A light force is rendered in the z direction, gently forcing the Phantom tip toward the nearest plane. The user can easily change from plane to plane, and thus from tool to tool, with a push strong enough to overcome this force. Additionally, stiff walls are rendered at the front and back of the set of planes.

The overall effect of these forces can be described as a series of haptic *detents* in the z direction, similar to the detents on some electronic device knobs or mouse scroll wheels, with hard stops at both ends. A tool is selected by moving the Phantom tip to the corresponding plane/detent. Planes are spaced 10 mm apart.

Within each plane, the Phantom tip is constrained inside a square region with sides of 20 mm. The magnitude of the force which provides this constraint is given by the following piecewise linear function, graphed in figure 3.2:

$$f(x) = \begin{cases} 0 & x < 0 \\ k_b x & 0 \leq x < x_b \\ k_b \frac{(x_w - x)}{x_w - x_b} & x_b \leq x < x_w \\ k_w(x - x_w) & x \geq x_w \end{cases}$$

where x is the distance of penetration of the Phantom tip beyond the boundary. From penetration $x = 0$ to $x = x_b$, the user experiences an initial semi-stiff spring force with

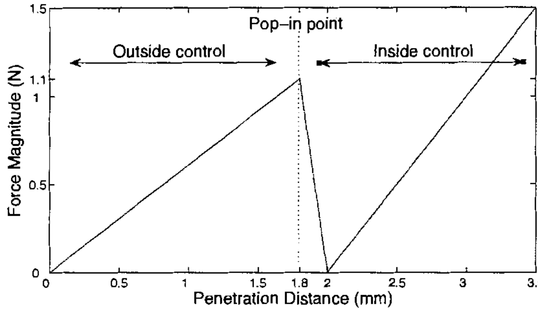


Figure 3.2: The magnitude of the force displayed at the constraint boundary as a function of penetration distance. Note that the *direction* of the force is always orthogonal to the wall, toward the interior of the square. In this example, $x_b = 1.8$ and $x_w = 2.0$. The tip pops into the boundary when the penetration distance exceeds x_b. The rightmost line rising from (2,0) theoretically extends to infinity, and represents the stiff wall encountered after popping in. (From [61]. Copyright 2006 ACM Press. Reprinted with permission.)

spring constant k_b. For penetration $x > x_w$, a second a second, stiffer spring force with constant k_w is displayed. Note that between x_b and x_w, the force tapers to zero. The resulting haptic effect is that, with a light push, the Phantom tip *pops* into a channel just outside the boundary, and with a light pull, it pops out of the channel, back into the interior of the square.

Each edge of the square is mapped to one or more parameters of the active tool. Edges mapped to more than one parameter are divided by small haptic bumps. Figure 3.3 shows a graphical display of the boundary for the text tool. The bottom edge of the boundary is divided into three segments, one each for bold, italic, and underline. The top edge is similarly divided into segments for left, right, and centre alignment.

A parameter is manipulated by popping into its boundary segment. For boolean parameters (such as *bold*) simply popping in toggles their value. For a parameter with a wider range of ordinal, nominal, or ratio values (such as *font size*), moving the Phantom tip along its boundary segment after popping in changes its values according to a specified segment-to-value mapping.

Parameters are manipulated in this fashion so that the user can haptically explore the constraint boundary without inadvertently changing a parameter's value. Values are only changed when the user exerts enough force on the boundary to pop in.

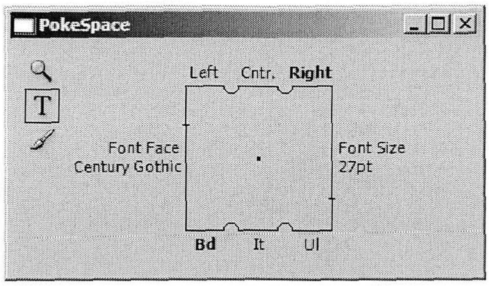


Figure 3.3: The Pokespace graphical display for the text tool. The small square in the center of the constraint boundary indicates the position of the Phantom tip. (From [61]. Copyright 2006 ACM Press. Reprinted with permission.)

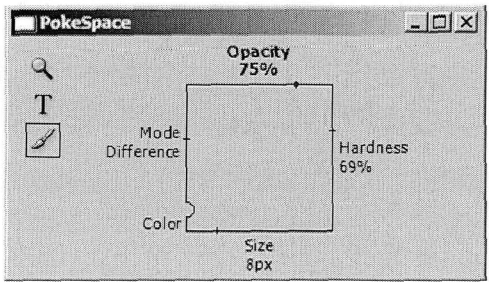


Figure 3.4: The Pokespace graphical display for the brush tool, while the brush's opacity parameter is being modified. The Phantom tip marker lies along the top edge of the constraint boundary. The parameter's label is also now displayed in boldface to indicate its active state.

3.3.2 Graphical display

The haptic rendering described above is accompanied by a graphical display similar in appearance to the mini-dialog windows found in several well-known design applications such as Adobe Photoshop.

Figure 3.3 shows the display with the text tool active. The icons on the left indicate the currently selected tool. The main part of the window shows a visualization of the constraint boundary, as described above, for the currently active tool. Each segment of the boundary is labeled with the name and current value of the parameter it controls. The labels of boolean parameters that are set to *true* are shown in boldface.

A blue marker dot displays the position of the Phantom tip in the plane. When the Phantom tip pops into a boundary segment of a non-boolean parameter, the label for that segment changes to boldface to indicate its active state. As the Phantom tip moves along

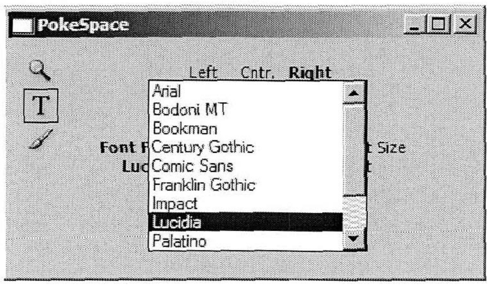


Figure 3.5: The Pokespace graphical display for the text tool, with the font face parameter's pop-up list box visible. The text box aids in the selection of non-boolean parameters with non-numerical value ranges or non-linear segment-to-value mappings. (From [61]. Copyright 2006 ACM Press. Reprinted with permission.)

the segment, the displayed value is continuously updated, much like a conventional slider. Figure 3.4 shows the graphical display for the brush tool as the opacity parameter is being modified.

For parameters with a non-numeric range or a non-linear segment-to-value mapping, the consequences of moving the Phantom tip a given distance may not be clear. In such cases, the display provides a pop-up list box containing the set of possible parameter values. The list box is displayed only while the parameter's segment is active. Selecting a value then becomes similar to using a standard dropdown list. Figure 3.5 shows such a pop-up list box for the text tool's font face parameter.

3.3.3 Additional haptic features

Beyond the basic haptic rendering, I experimented with several additional haptic features in an effort to enhance the haptic experience.

A light force attracting the Phantom tip to the center of the boundary square was rendered when no boundary segments were active. This force was light enough to permit free movement within the square, but strong enough to center the Phantom tip when allowed to drift freely. The re-centering force was intended to provide an eyes-free cue as to the relative position of the Phantom tip within the boundary square, much as the raised bumps on the F and J keys of a standard keyboard help users re-orient their hands without looking.

A small haptic detent was rendered in a non-boolean parameter's boundary segment at the position corresponding to that parameter's current value. This was intended to allow the user to more easily pop into the segment at that position when small adjustments to the current value were desired.

For non-boolean parameters with a relatively small number of possible values, such as the brush tool's *mode* parameter, a series of detents was rendered along the direction of the parameter's segment, with one detent for each possible value. The detents were only rendered when the parameter was active. This effect was intended to provide a concrete haptic cue that the parameter value had changed, again similar to some knobs.

For all other non-boolean parameters, a light friction force along the direction of the segment was rendered when the parameter was active, using the friction model of Ho, Basdogan, and Srinivasan [24]. The friction force was intended to stabilize the user's movement along the segment, much as the friction of a table top does for mouse movements.

Only the center attractive force was sufficiently effective to be included in the studied design. The haptic detents along the boundary proved problematic as it was difficult to move to the neighboring detent without skipping it. This may be due to the rendering algorithm or to limitations of the haptic device. The friction force was neither problematic nor helpful, so it was left out.

3.3.4 Application of Design Principles

This section reviews the guidelines set down in section 3.1.2 and examines how Pokespace follows them.

Design for rehearsal. Pokespace supports rehearsal through its provision of a visual display. Novice users can guide their movements using the graphical display, while still experiencing the haptic feedback associated with the action. With practice, it is hoped, reliance on the graphical display will diminish, and haptic feedback will become sufficient.

Design for gestures. There are several sequences of actions in Pokespace which may be considered to be gestures, especially those taking advantage of backstops. A good example is the action of switching to the tool plane closest to the user, followed by popping into one of the corners of the boundary square, thereby setting one of the parameters to its minimum or maximum value. That action makes use of the front z-wall and two sides of the boundary square to assist the movement, and thus seems likely to develop into an efficient gesture.

Keep the haptic environment simple and consistent. I feel that the haptic layout of Pokespace, both within a plane and along the z axis, is sufficiently consistent and simple to support motor memory. While individual settings for some parameters are not likely to be remembered, the adjustment of many such settings (e.g. font size or zoom level) can be monitored without diverting attention from the object of interest.

Maximize the number of backstops. Pokespace contains many backstops, including the front and back walls, the planar boundaries, and the lateral stops at the ends of a boundary segment.

Promote visual attention on the object of interest. I feel that Pokespace certainly eliminates the need for visual attention to the interface controls in many cases. Moreover, it degrades well — for those cases where a glance is still required, the graphical display still provides a clear depiction of the interaction.

Support integrality of movement. Both the z direction detent forces and the planar boundary forces serve this purpose. However, it should be noted that in popping out of a boundary segment, the user is required to restrict their movements orthogonally to the segment, or risk inadvertently changing the parameter value in the process. This represents a deviation from the integrality guideline, which must be addressed in future designs.

Make optimal use of the device's degrees-of-freedom. The use of the z dimension in Pokespace is particularly effective since it exploits the third DOF without significantly sacrificing the clarity of the graphical display — a simple stack metaphor is all that the user must learn.

Use the non-dominant hand for haptic feedback. This guideline is clearly followed.

3.3.5 Usage Scenarios

The task of drawing a moustache on a digital photograph of a friend (or foe) using a typical image editing application offers some concrete examples of the potential benefits of Pokespace:

In order to create a moustache of just the right thickness and texture, the user may want to fine tune the size, hardness, and opacity parameters of the brush tool through repeated experimentation. With the conventional interaction technique, this would require many mouse movements between the canvas and the parameter controls, constantly testing one setting after another. With Pokespace, a well practiced user could start drawing a test stroke with the mouse and alternately manipulate several brush parameters with the Phantom while keeping their attention on the canvas as the parameters are changing.

In the course of drawing the moustache, the user is bound to make mistakes which require fixing with the eraser tool. Note that a typical eraser tool has similar parameters to a brush. If the eraser tool's plane were nearby (ideally neighboring) the brush tool's plane, the user would quickly, through rehearsal, become accustomed to the movements between the two tools and into and out of their respective parameters. This cannot be said for the mouse movements between the canvas and a conventional tool palette, and between the palette and the parameter controls.

One of the most commonly used tools in many design programs is the selection tool. For example, in drawing the moustache, the user may wish to select a mask region to avoid accidentally painting the person's nose or lips with the brush tool, then select a different region to trim with the eraser, then select a third region to fill with the paint bucket. Common tools like the selection tool could be placed at the extreme front or rear walls of the Pokespace, so that activating them would require an imprecise forward or backward thrust with the Phantom tip. While the movements to each of the brush, eraser, and paint bucket tools are slightly different, the movement back to the selection tool is effectively the same regardless of starting location, and could easily be performed without looking.

3.4 Conclusion

This chapter began by making the claim that despite its limitations, point force haptic feedback still has the potential to enable more effective interaction techniques. Hopefully, the design principles and the Pokespace interaction technique presented in this chapter have begun to convince the reader of the validity of this claim. Of course, empirical data is necessary to validate both the fitness of the principles, and the effectiveness of Pokespace itself. Initial efforts in that direction are presented in the next chapters.

This chapter was also intended to describe the line of reasoning that led to the present work. I suggest that some of those ideas are independently valid, and I hope that they will be considered by other researchers involved in similar work, perhaps in designs radically different from Pokespace.

Chapter 4

Study 1: Basic Effects of Haptic Feedback

As discussed in the previous chapters, the concept of a bimanual interaction technique involving haptic feedback seems promising. However, it is not well studied — to my knowledge, only two groups have developed such techniques [4] [17]. One of those groups reported negative results, and the other reported none at all.

Therefore, many questions exist about the utility of haptic feedback in a bimanual interaction technique:

- Will users find haptic feedback helpful, or will it just add confusion to an already complex situation?
- If it is helpful, what is helpful about it?
- Can it encourage simultaneous, coordinated movement in both hands?
- Can it reduce the user's mental workload?

In an effort to address some of these questions, the study described in this chapter explored the basic effects of haptic feedback on a simple combined selecting and positioning task. As per the design guidelines of the previous chapter, the mouse was held in the dominant hand and performed the positioning task, while the Phantom was held in the non-dominant hand and performed the selecting task. The modality through which feedback was provided for the selecting task varied between visual and haptic. I expected that the haptic condition would support both faster performance and lower workload, and encourage higher bimanual coordination.

4.1 Methods

4.1.1 Task

The experimental display is shown in Figure 4.1. At the start of each trial, a small circle containing a number from 1 to 8 appeared at a random location in the white area of the display. Participants were required to select the appropriate number with their non-dominant hand using the Phantom, and to point to and click within the circle with their dominant hand using the mouse. The dominant hand task was the same for all conditions, while the non-dominant hand Phantom movement varied between three conditions.

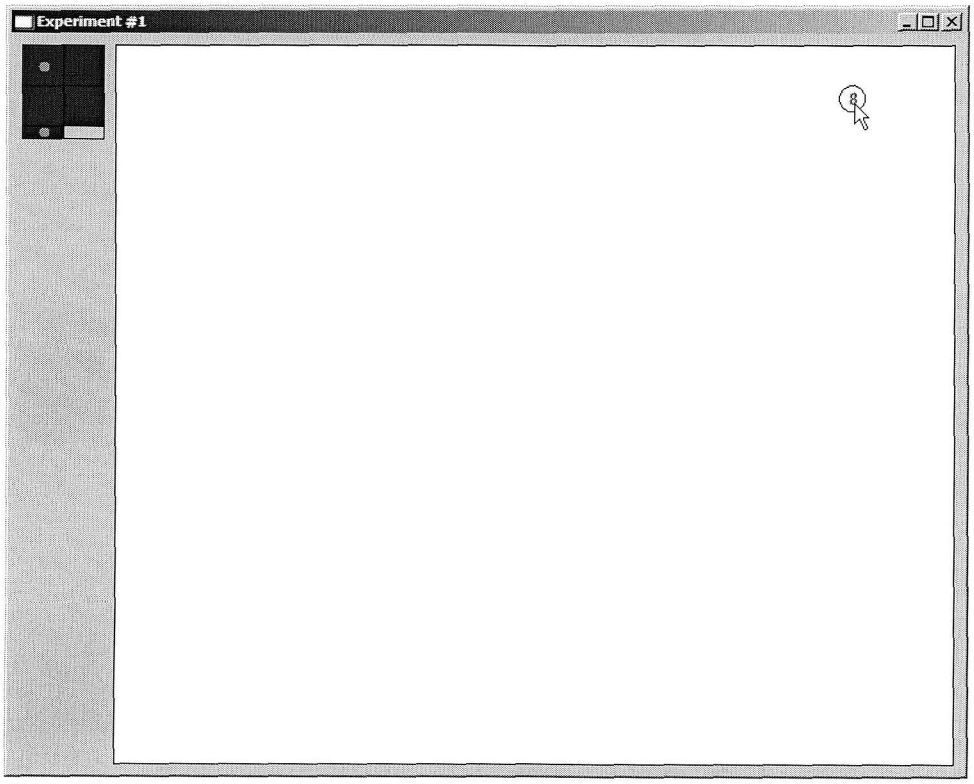


Figure 4.1: The experimental display for Study 1.

The working volume of the Phantom was divided into octants about the x, y, and z axes. Octants were assigned the numbers 1 to 8. To select a given number, participants placed the Phantom stylus tip anywhere within the corresponding octant. The selecting task was organized in this fashion to make it as easy as possible to select a given number in a three-dimensional space.

The experiment compared three variations of this number selecting task. The first condition was named force feedback (F), in which the user's movements were restricted to the interior of a virtual cube rendered by the Phantom. The cube had a width of 2 cm, and was centered about the origin of the space described above.

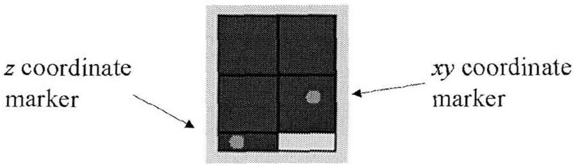


Figure 4.2: The graphical Phantom position display provided in the GA and GR conditions. The upper marker indicated the x and y coordinates of the Phantom, while the lower marker indicated its z coordinate.

In the second condition, the haptic rendering of the cube was removed. Instead, participants were issued visual feedback of their position within the cube with a small display (Figure 4.2). The upper marker indicated their x and y coordinates, while the lower marker indicated their z coordinate. If participants moved outside the cube in any direction, the corresponding marker stopped at the edge of the indicator, just as a conventional mouse pointer stops at the edge of the screen. The background colour of the indicator also changed from dark gray to light gray when the tip crossed the plane $z = 0$. This condition is referred to as graphical-absolute, and denoted GA.

In both the F and GA conditions, the origin of the octant system was fixed to the origin of the Phantom's working volume. In the third condition, called graphical-relative (GR), the octant system origin shifted whenever participants moved the tip against the bounds of the cube. In other words, participants dragged the cube around with the tip. This is similar to the usual behaviour of a mouse, whereby the origin of the screen coordinate system shifts with respect to the physical table coordinate system whenever the pointer moves against the edge of the screen.

While the difference between the GA and GR conditions is slight, it was not clear which would support better performance. The GA technique was the direct non-haptic analog of the F technique, while the GR technique was more closely related to the familiar behaviour of the mouse. Therefore, both were included.

No graphical feedback was issued in F, and octants were not labeled in GA or GR, in order to bring about skilled performance in participants with as little practice as possible. Pilot studies in which octants were labeled, and in which graphical feedback was issued in all conditions, showed that participants would tend to use the graphical display as a crutch instead of learning the numberings or becoming habituated with the haptic feedback.

I acknowledge that there may exist non-haptic or unimanual interaction techniques which support better performance for the particular task used in this study. These techniques were not designed primarily for optimal performance, but to investigate the effects of the addition of haptic feedback to a bimanual technique.

4.1.2 Experimental design

A mixed design consisting of one within-subjects effect (force feedback vs. graphical feedback) and one between-subjects effect (relative vs. absolute) was chosen. Half the participants were tested under the F and GA conditions, and the other half under the F and GR conditions. The order of the F and G conditions was counterbalanced.

Hypotheses. I expected that participants would perform faster in the haptic condition, chiefly because the haptic feedback would support simultaneous movement in both hands. I also expected that the haptic condition would result in a lower subjective workload, since participants would not have to shift their visual attention as often as in the graphical condition.

Participants. Participants were 12 right-handed (self-reported) volunteers, (10 male, 2 female), aged 23 to 42 years, with a median age of 25. All were computing science graduate students at Simon Fraser University. Participants were recruited by email solicitation, and were rewarded for their participation with baked goods.

Protocol. Participants first read a written instruction sheet summarizing the experiment, and completed a consent form and background questionnaire.

Next, participants were introduced to the Phantom. Only 2 reported having used such a device more than once or twice. A simple demo program which renders a frictionless sphere was run, and participants were asked to poke at the sphere with the stylus. Subjects were encouraged not to be afraid of breaking the Phantom, despite its fragile appearance.

Once apparently comfortable with the Phantom, verbal instructions were issued to participants. An oversized physical model was used to acquaint participants with the numbering of the octants.

Participants completed 1 practice block plus 5 timed blocks of 25 trials for each condition. Practice trials were monitored by the experimenter, and any abnormal or dangerous behaviour, such as pushing too hard on the Phantom stylus, was pointed out to participants at that time. Optional 20 second breaks were allowed between trials. Time data for the first trial of each block was discarded to allow for participants' readjustment to the apparatus after taking a break. Thus, for analysis each block contained 24 trials.

NASA TLX workload questionnaires were completed after each condition, and an open-ended subjective questionnaire was completed at the end. Sessions lasted approximately 35 to 45 minutes.

4.1.3 Apparatus

The haptic device used was a Phantom Premium 1.0, driven by a dual Intel Xeon 3.06 GHz system with 2 GB of RAM at a haptic refresh rate of 1000 Hz. The mouse was a Microsoft IntelliMouse Optical 1.1A. In the Windows Control Panel, the mouse speed was set to 5 on a scale from 0 to 10, and mouse acceleration was turned on. Mouse and Phantom path data were recorded at a rate of 100 Hz. The experiment's interface was developed using Microsoft Visual C++ .NET 2003, and ran on Microsoft Windows XP 2002 SP2. No other applications were running during the experiment.

4.2 Results

Analysis of the data revealed no significant differences for the GA and GR conditions for any dependent variable. In addition, observation of participants revealed no apparent difference between the conditions in strategy or usage habits. Therefore, in the remaining discussion, the two conditions are grouped into a single graphical feedback condition called G.

4.2.1 Indications of skilled performance

Error counts and mean block times indicated that the participants quickly reached skilled performance. Means of completion times for the first six blocks performed by each participant were compared to assess performance improvement over time. A plot of the means across each block for both conditions is shown in Figure 4.3. A clear improvement is seen between block 1 (the practice block) and block 2 (the first timed block). No appreciable improvement subsequently occurs. This indicates that participants became skilled during the initial 25 practice trials in both conditions. Consequently, the 120 trials in blocks 2–6 were used in the analysis of the time results.

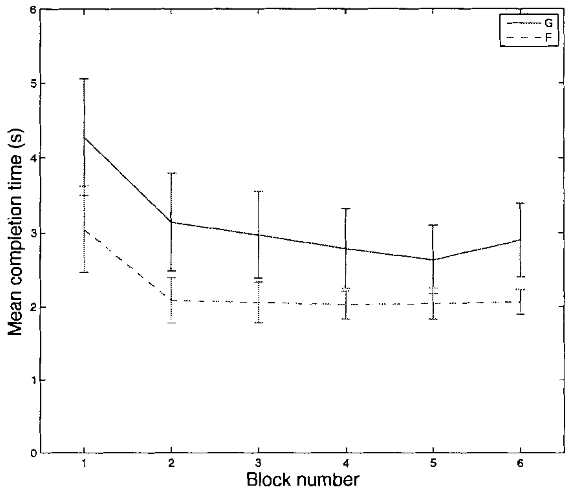


Figure 4.3: Mean completion times per block for the first six blocks performed by all 12 participants. Error bars show ± one standard deviation. While a clear improvement is seen between block 1 (the practice block) and block 2 (the first timed block), no appreciable improvement is seen thereafter.

Error counts were low, further suggesting that participants quickly became skilled. Two types of errors were recorded. *Number selection errors* occurred when participants clicked within the circle, but had selected the wrong number. *Targeting errors* occurred when participants failed to click within the circle, regardless of the currently selected number. The following analysis does not include errors occurring during practice trials.

Mean error percentages were 4.2% (SD = 3.7%) for number selection and 3.5% (SD = 3.6%) for targeting for the F condition, and 7.4% (SD = 5.7%) and 2.5% (SD = 2.8%) for the G condition. Overall, these indicate that the users had little problem recalling the correct octant to select a number, although other factors appear to have increased the difficulty in the G condition.

All but 3 participants performed more number selection errors in the G condition, while all but 2 performed more targeting errors in the F condition. However, differences in number selection errors were more pronounced.

The differences in errors between the F and G conditions were small, at 3.92 ± 5.28

Table 4.1: Subjective workloads for each participant for F and G conditions. Only participants 1 and 10 reported markedly higher workloads for the F condition.

Condition	1	2	3	4	5	6
G	2.7	10.7	15.9	8.2	11.9	11.5
F	6.1	11.2	14.0	3.9	11.7	9.4

Condition	7	8	9	10	11	12
G	13.5	13.7	7.8	3.6	14.7	6.5
F	8.2	9.4	5.1	7.8	7.3	5.5

(95% confidence, $p = .13$) number selection errors and 1.16 ± 1.48 targeting errors (95% confidence, $p = .11$).

No appreciable differences were observed between mean error counts for the first (non-practice) and last blocks performed by participants in either condition.

Together, these data suggest performance was consistent across blocks 2–6. For the remainder of this analysis, it is assumed that this was the case.

4.2.2 Completion time

Quantile-normal plots of trial times revealed that the time data was non-normal, but log-transformed data was approximately normal. Thus, log-transformed values are used for the remainder of the analysis.

Paired-comparison t-tests performed on the means of the log completion times for each participant revealed that the F condition was significantly faster than G ($t(11) = 5.94, p < .001$), by about 25% (95% CI = [15%,36%]).

4.2.3 Workload

Workload data was collected using the NASA TLX instrument [21]. A paired t-test revealed marginally significant workload differences between F and G conditions ($t(11) = 1.80, p = .099$).

Table 4.1 shows the calculated workloads for each participant. Notably, the only participants who reported markedly higher workloads for the F condition (participants 1 and 10) were the only two participants who complained of arm fatigue in their subjective questionnaires. This suggests that their reported difference in workload may have been due to discomfort with the device itself rather than the interaction technique. Care should thus be taken in future to mitigate such fatigue.

4.2.4 Path data

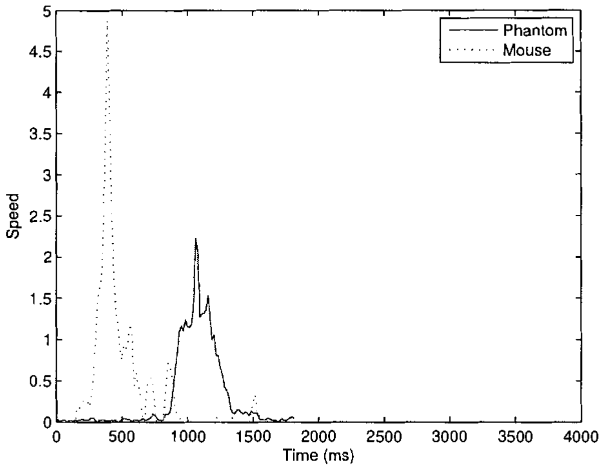


Figure 4.4: Mouse and Phantom speeds for trial 2 of block 6 in the F condition by participant 5. The mouse clearly completed its movement before the Phantom began.

Based on the recorded path data, the speeds of both the mouse and the Phantom were calculated using the difference in Euclidean distance from one timestep to the next. Those speeds were then plotted for each trial from a sample of blocks from each participant. Speeds for the Phantom (in cm/ms) were scaled by a factor of 10 for the purposes of comparison to mouse speeds (in pixels/ms). A typical example of such a plot is given in Figure 4.4.

The first sharp spike in Figure 4.4 depicts a fast mouse movement at the start of the trial. At the tail end of that spike, a slightly wider hump shows a movement of the Phantom. Some small, presumably corrective movements of the mouse then come near the end of the Phantom's hump, and the trial ends. The mouse and Phantom movements exhibit little simultaneity. Figure 4.5 is similar to Figure 4.4, except for the two distinct maxima in the Phantom's path, which likely represent a corrective movement following an error. Here also, little simultaneity is evident.

Comparison of the times at which the maximum speeds of the mouse and Phantom were achieved supported the observation that the mouse moved first in most trials. Paired t-tests revealed significant differences in times of peak speeds for the mouse and Phantom for both

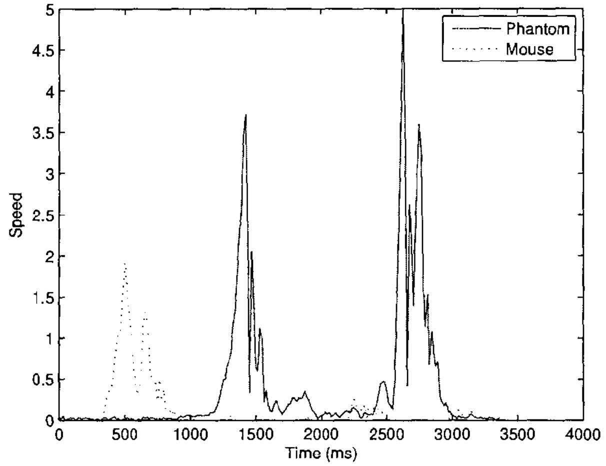


Figure 4.5: Mouse and Phantom speeds for trial 11 of block 6 in the F condition by participant 5. Two spikes are evident in the path of the Phantom, likely indicating an error correction.

the G condition ($t(11) = 5.39, p < 0.001$, 95% CI [380.8 ms, 906.2 ms]) and the F condition ($t(11) = 11.23, p < 0.001$, 95% CI [530.7 ms, 789.4 ms]).

To derive a measure of the degree of simultaneity between the movements of the mouse and the Phantom, the similarity of their speed functions was estimated. Specifically,

$$\frac{\mathbf{f} \cdot \mathbf{g}}{\|\mathbf{f}\|\|\mathbf{g}\|}$$

was computed where f and g are the speed functions of the mouse and Phantom, treated as vectors. This resulted in a value between 0 and 1, where 0 implies no simultaneity, and 1 implies total simultaneity. Simultaneity as estimated by this measure was generally low, at an average of .27 for the G condition and .22 for the F condition.

Another notable trend in path data is suggested by Figure 4.6, which shows a typical trial in the G condition. The same sharp mouse movement is evident at the start of the trial. The following Phantom movement is also present, but takes place over a longer period. This suggests that participants moved the Phantom at similar speeds but over longer distances in this condition.

The data support this suggestion. A comparison of the mean distance moved by the Phantom in each condition revealed significantly greater distances in the G condition than in the F condition ($t(11) = 2.23, p = .047$, 95% CI [0.10 mm, 12.77 mm]).

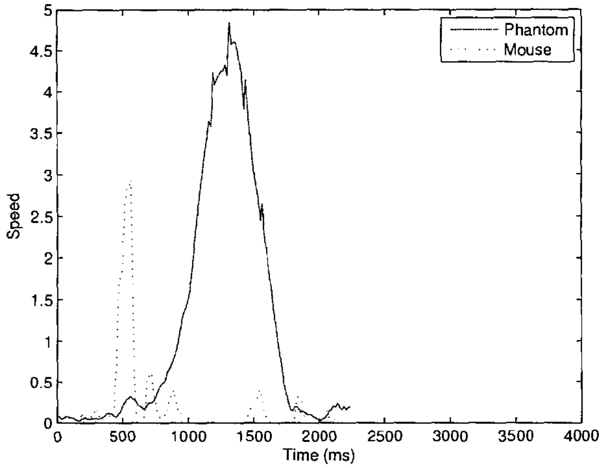


Figure 4.6: Mouse and Phantom speeds for trial 9 of block 6 in the G condition by participant 5. The Phantom's spike is considerably wider than in Figure 4.4, indicating that the Phantom moved a greater distance.

Despite the trends described above, a non-trivial amount of individual difference was observed across participants. In particular, the amount of simultaneity was noticeably greater for at least three participants. However, increased simultaneity did not appear to affect completion times — those three participants exhibited near-average performance in both conditions, with the exception of one who was slower than the average for the G condition.

4.2.5 Qualitative Results

In subjective questionnaires, 11 out of 12 participants said that they preferred the F condition, claiming that it provided better awareness of their current position in the octant space (especially in the z direction), allowed them to focus their attention more on clicking the circle, and required less arm movement than the G condition. One participant preferred the F condition "since it restricts the range of motion. The force feedback frees the eyes from tracking the location of the [Phantom]". This opinion was common.

Most participants were very receptive to the Phantom device, and quickly became accustomed to moving within the cube. However, a small number of participants claimed that the cube was "too small". This was puzzling, since it was assumed that participants would determine their position in the cube by pushing against the walls. In this case, the size of the cube should have little effect on the perceived utility of the feedback. The participants in question may not have adopted that strategy.

Several participants reported having not made much use of the graphical feedback in the GA/GR condition. One participant commented, "For the most part I did not use the [graphical Phantom position display] at all, but I was using the [background color] changes from the corner of my eye." Another reported that rather than shift his gaze continually from the graphical Phantom position display to the target, he ended up making "wider, more deliberate gestures." Overall, it appears most participants checked the graphical Phantom position display from time to time, but not on every trial.

4.3 Discussion

The two most striking results of this study were the faster completion times in the F condition and the lack of bimanual simultaneity of movement. Each results warrants discussion.

I suggest that the reasons for faster performance in the F conditions are two-fold. First, as shown, in the G conditions, participants wound up moving the Phantom further distances, naturally leading to longer completion times. Lacking the backstop provided by the F condition, users were forced into a trade-off: if they wished to save the time required to shift their gaze to the graphical Phantom position display, they had to move the Phantom a greater distance to ensure they had selected the proper number.

Second, the F condition may also have reduced the time spent in the final phase of movement: verification that the intended target has been reached [66]. While the graphical Phantom position display did provide such confirmation, participants had to shift their gaze to see it, whereas the haptic sensation of the corners of the cube was available immediately. Also, as stated in chapter 2, reaction time to haptic cues has been shown to be faster than to visual cues in some situations. Thus the movement verification phase was likely shorter for the F condition.

Beyond the specific task used in this study, the faster performance in the F condition suggests several potential benefits of haptic feedback for interaction techniques. First and most simply, it suggests that users were able to perceive the haptic stimuli, and that the feedback was not disorienting or confusing. Second, since the F condition had no graphical feedback for the selecting task, it suggests that haptic feedback has the potential to allow increased visual attention on the object of interest, as per Design Guideline 5. Finally, it confirms the potential for haptic backstops to shorten movement distance and provide cues that a movement has reached its target, as recommended in Design Guideline 4.

As stated, participants reported making only limited use of the graphical Phantom position display in the G condition. This seems to agree with the finding that participants moved the Phantom further in those conditions. If the display had been attended to on every trial, participants would not presumably have needed to make "wider, more deliberate gestures" in order to ensure that the correct number was selected. Further study involving eye tracking data would be required to conclusively confirm this suspicion.

Analysis of the path data showed that a relatively small amount of bimanual coordination was exhibited by participants in any of the conditions. This was puzzling given the prevalence of coordinated movement in everyday life.

The explanation is most likely that users had insufficient practice to develop simultaneity. The bimanual coordination of asymmetric movements observed in everyday life, such as the ability to shift the gears, press the clutch, and turn the steering wheel of an automobile simultaneously, are only developed after considerable practice.

Note that while it is argued above that performance was stable throughout most of the experiment, this does not preclude the possibility that coordination would have developed with more practice. Still, this result has demonstrated that bimanual coordination should not be taken for granted, even if conditions seem appropriate for it to develop. Many factors should be considered when designing for coordination, including the symmetry of the action [58], the familiarity of the actions of each hand, and the expected frequency of usage of the interface.

The path data also showed that participants moved the mouse before the Phantom in most trials, and that they usually did not move the Phantom until the mouse movement was almost complete. This was also not expected. Given that the Phantom's task must be completed before that of the mouse, the natural order would seem to be Phantom-first. Instead, the tendency in most cases was to first make a sharp movement with the mouse toward the target, then move the Phantom to the specified octant, and finally move the mouse the remaining distance to the target and click.

A possible explanation for this behaviour is the difference in familiarity of users with the mouse compared with the Phantom. In particular, all participants had ample experience with mouse pointing, and thus the apparition of a target to be clicked likely provoked a practiced response. Meanwhile, the action of pointing at an octant in three-space would be much less familiar to most participants. Thus the action of moving the mouse was available more quickly, and was performed first.

Another possible contributor to this behaviour is that participants had to look at the dominant hand target, in order to determine the octant number, before the non-dominant hand movement could begin. Therefore, the natural tendency would seem to be to initiate the dominant hand movement first, since its target was already being attended to. It also appears that participants used the time during which the initial sharp dominant hand movement was being executed to start planning the non-dominant hand movement, since the latter usually began immediately after the former, or even as the former was finishing. These behaviours highlight the adaptability, ingenuity, and unpredictability of humans in developing strategies for interaction tasks. While not directly quantifiable, that notion constitutes one of the major lessons learned from this study.

Those lessons contributed significantly to the design of Pokespace, the evaluation of which is the subject of the next chapter.

Chapter 5

Studies 2 and 3: Initial Evaluation of Pokespace

The second and third studies presented in this thesis investigated Pokespace directly, in order to gain insight into the strengths and weaknesses of the design, and to answer questions such as the following:

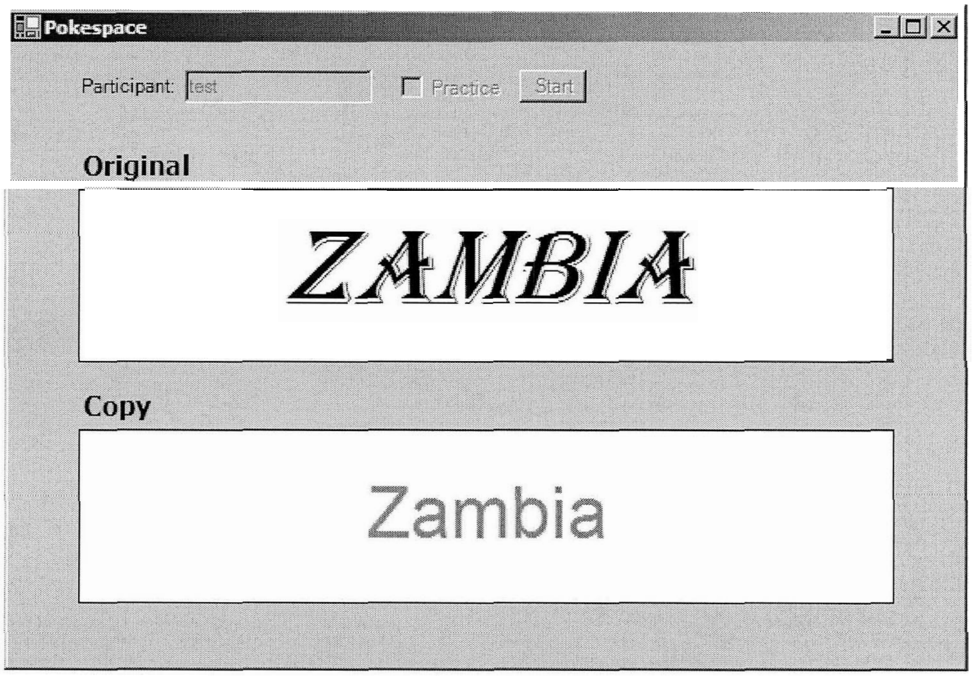
- Will users tend to ignore visual feedback once haptic feedback becomes familiar? What if we force them to rely on haptic feedback?
- Will users learn Pokespace easily? Will performance continue to improve over time?
- How will Pokespace compare to an interface composed of traditional GUI widgets?
- Will the multimodal feedback be comfortable or overwhelming?

For the sake of simplicity, the implementation of Pokespace used in these first studies consisted of only a single tool, the text tool. Also, no task was performed with the dominant hand. Future studies will consider multiple tools (requiring multiple planes) and tasks requiring the combined use of both hands.

5.1 Overview

This section describes the characteristics common to both studies described in this chapter.

Portions previously published [61]. Copyright 2006 ACM Press. Adapted with permission.



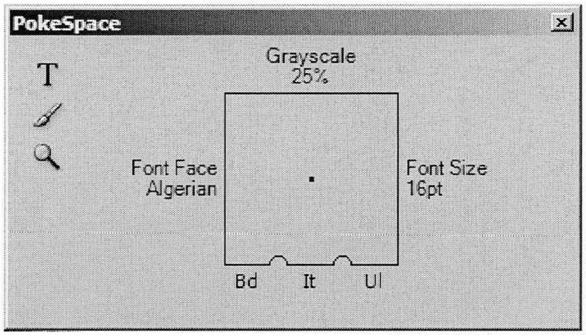
Participant: test

Figure 5.1: The window displaying the words in the text matching task. Participants had to modify the formatting of the bottom word to match the top one. (From [61]. Copyright 2006 ACM Press. Reprinted with permission.)

5.1.1 Task

Participants were shown two copies of the same word, one underneath the other. A screenshot of the word display is shown in figure 5.1. The words were chosen randomly from a list of countries. At the start of each trial, the bottom word was displayed in 50% gray 32 pt plain Arial font. The top word was displayed in one of four fonts (Algerian, Comic Sans MS, Stencil, and Times New Roman), chosen to be easily distinguished from each other and Arial; with size 16 pt, 24 pt, 40 pt, 48 pt, or 56 pt; in 25%, 75%, or 100% gray; and exactly one of bold, italicized, or underlined. The properties for a given trial were randomly selected from these sets. Participants were asked to modify the bottom word's properties so that it typographically matched the top word, and then to press the 'N' key on the keyboard, advancing to the next trial.

Modification of the word's properties was accomplished with two different interaction techniques. The first was a simplified version of Pokespace in which there was only one tool (a single plane). The graphical display showing the configuration of Pokespace for



PokeSpace

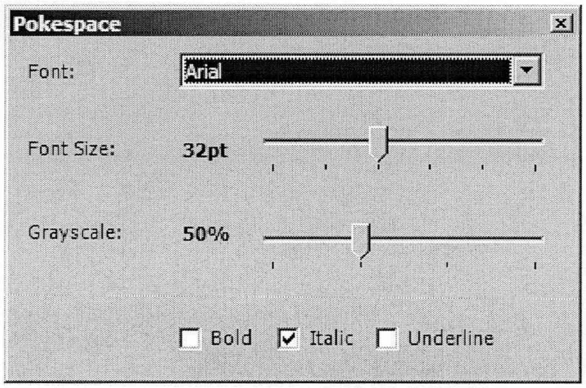
Grayscale
25%

Font Face
Algerian

Font Size
16pt

Bd It Ul

Figure 5.2: The Pokespace graphical display used in Studies 2 and 3. Only the text tool was implemented.



Pokespace

Font: Arial

Font Size: 32pt

Grayscale: 50%

☐ Bold ☑ Italic ☐ Underline

Figure 5.3: The traditional GUI controls used in Study 3. A combo box controls the font, sliders control the font size and grayscale, and checkboxes control the bold, italic, and underline. (From [61]. Copyright 2006 ACM Press. Reprinted with permission.)

the studies is shown in Figure 5.2, with the three controls on the top boundary segment replaced with a single control for grayscale. For this technique, the Phantom was held in the participant's non-dominant hand, as per the intended use of Pokespace.

The second technique consisted of a mini-dialog window containing a set of traditional GUI controls—one combo box, two sliders, and three check boxes. These controls are shown in figure 5.3. Participants operated the controls with their dominant hand using the mouse, while the non-dominant hand completed the trial by pressing 'N'.

The graphical display for Pokespace and the traditional control dialog window were located at the top left of the screen, while the word pair was displayed at the center of the screen. The distance from the center of the word display to the center of the Pokespace display or traditional controls was about 625 pixels or 15 cm.

5.1.2 Apparatus

The studies were run on a machine with dual Intel Xeon 3.06 GHz processors and 2 GB of RAM, running Microsoft Windows XP 2002 SP2. The haptic device was a Phantom Premium 1.0, driven at a refresh rate of 1000 Hz. The mouse was a Microsoft IntelliMouse Optical 1.1A. Pokespace and the experimental interface were developed using Microsoft Visual C++ .NET 2003. The display was a 17" TFT LCD monitor with a resolution of 1280x1024. A Tobii 1750 eye tracker and Tobii Clearview software ran on the same machine, collecting the gaze data and recording screen contents. No other applications were running during the experiment. A stack of books was placed in front of the Phantom to provide an arm rest for participants.

5.2 Study 2

Study 2 was designed to identify and examine trends in usage of Pokespace over a moderately long period (approximately 30 minutes).

Design. Study 2 had only one condition, using Pokespace for the text matching task. Trial times and eye gaze coordinates were recorded, and users completed subjective evaluations and background questionnaires.

Hypotheses. I expected that participants would learn to rely on haptic feedback for monitoring the modification of parameters, so that eye gaze would seldom stray from the words in the centre of the screen. I also expected response time to improve dramatically during the first two blocks, after which improvement would continue at a lower rate.

Participants. Study 2 had 6 participants (5 male, 1 female), aged 23 to 27 years, with a median age of 24.5. All participants were graduate students from the schools of computing science and engineering science at Simon Fraser University. No participant had more than trivial experience with haptic interfaces. All participants reported using a computer 14 or more hours per week. Participants were each paid CDN$20.

Protocol. Participants first read a written instruction sheet summarizing the experiment, and completed a consent form.

Next, the workspace was adjusted to the participant, and the eye tracker was calibrated (participants were thus aware that their eyes were being tracked). Participants were then introduced to the Phantom. A simple demo program which graphically and haptically renders a set of polyhedra was run, and participants were shown how to grip the Phantom stylus and poke at the objects.

Once apparently comfortable with the Phantom, verbal instructions were issued to participants describing the task and the interaction technique. Thirteen blocks of 10 trials were then completed. There were no practice trials. Short breaks were allowed between blocks.

The subjective evaluations and background questionnaires were completed at the end of the session. Sessions lasted approximately 50 minutes.

5.2.1 Results

The Tobii ClearView software's fixation detection algorithm was used to analyze gaze fixations with a 30 pixel fixation radius and a 100 ms minimum fixation duration. Two regions were defined: one containing the word display in the center of the screen, and one containing the Pokespace graphical display in the top left corner.

Participants usually fixated several times consecutively in one region before shifting their gaze to the other. We were primarily interested not in how many individual fixations occurred, but how often participants' gaze shifted to the Pokespace display. We therefore defined a *glance* as an incidence of one fixation on the word display immediately followed by a fixation on the Pokespace display.

Data for one participant was improperly recorded and had to be discarded. A graph of each of the remaining 5 participants' glance counts per block is shown in figure 5.4. While 1 participant made almost no glances after the second block, the other 4 participants glanced regularly throughout the session.

Of the 4 who continued to glance at Pokespace, 1 glanced markedly less. Examination of the screen recordings for that participant revealed that most of their glances occurred when the bold, italic, or underline parameters were modified.

A graph of mean completion times per block for each participant is shown in figure 5.5. Mean completion times per block improved sharply for the first 3 to 4 blocks, and then began to level off, reaching approximately 11 s per trial by block 6.

Three issues with Pokespace's design were identified by participants in their subjective questionnaires. First, several participants stated that the hardest controls to manipulate eyes-free were the bold/italic/underline controls, and 1 of those stated that this was because they found it hard to know when the Phantom tip was in the centre of the square. This agrees with the gaze data described above. Second, several participants were distracted by the sudden change in parameter value that occurs when the Phantom tip first pops in to a

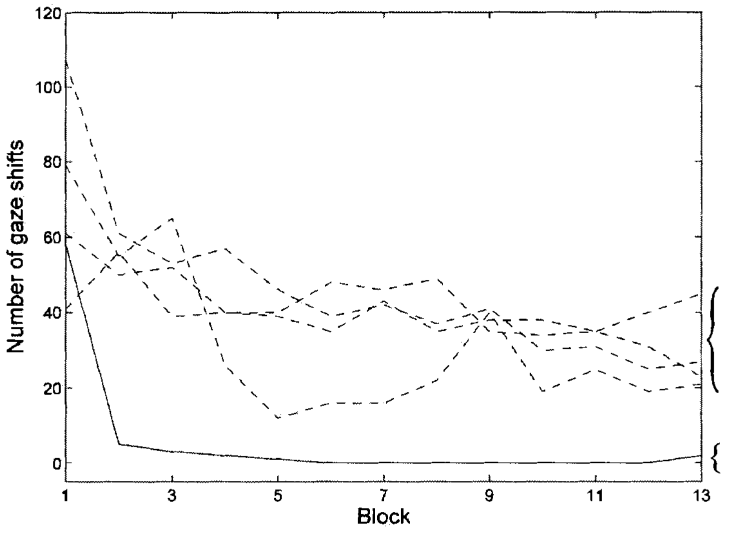


Figure 5.4: Glance counts per block for participants in Study 2 (data for one participant was improperly recorded and had to be discarded). One participant almost never glanced after the first few blocks, while the other participants glanced frequently throughout the session. (From [61]. Copyright 2006 ACM Press. Reprinted with permission.)

boundary segment. Finally, several participants stated that they were prone to inadvertently change a parameter as they pulled the Phantom tip away from the boundary.

5.3 Study 3

Study 3 compared Pokespace to a variant with no graphical feedback, and to the traditional interaction technique described above.

Design. Study 3 used a single-factor within-subjects design, in which interaction technique was a factor with three conditions. The first condition used standard Pokespace with graphical display (PSG); the second used Pokespace, but with no graphical display (PSNG); and the third condition used the traditional GUI controls (TR) described above.

PSNG had no practice period; instead it was always performed directly after PSG. While this was a non-standard experimental design, I felt that it allowed the most direct measurement of the effect of removing the graphical display, and ensured participants had enough practice with the combined haptic/graphic display (30 trials) to allow the transition to pure haptic use. The order of PSG/PSNG and TR was counterbalanced.

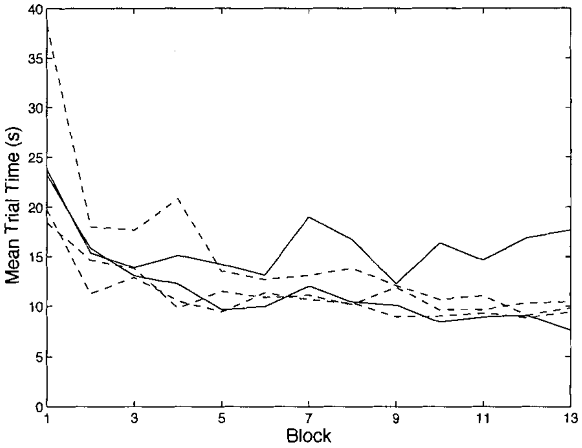


Figure 5.5: Mean completion times per block for participants in Study 2. Each line represents one participant. Different line types (dashed vs. solid) are provided for contrast. Performance improves sharply for the first 3 to 4 blocks, and then begins to level off at about 11 s per trial.

Trial times and eye gaze coordinates were recorded, and participants completed NASA TLX workload questionnaires, open-ended subjective questionnaires, and background questionnaires.

Hypotheses. I expected participants to shift their gaze to the graphical display far more frequently for TR than for PSG. As a result, I predicted that PSG times would be faster than TR times. I also expected that after performing the technique over 30 times in the PSG condition, participants would have little difficulty when the graphical display was removed, making performance times for PSNG similar to or faster than PSG.

I expected no significant difference between subjective workloads for PSG and PSNG but, given participants' familiarity with traditional interfaces, expected that workloads for TR would be slightly lower than for PSG.

Participants. Study 3 had 12 participants (10 male, 2 female), aged 23 to 29 years, with a median age of 24.5. No participants had performed in Study 2. All participants were graduate students from the schools of computing science and engineering science at Simon Fraser University. No participant had more than trivial experience with haptic interfaces. Eleven participants reported using a computer 14 or more hours per week, and one reported between 7 and 14 hours. Participants were each paid CDN$20.

Protocol. The instructions, calibration, and workspace adjustments were the same as in Study 2.

For the PSG and TR conditions, participants first completed five minutes of practice. In this time, participants completed an average of 17 trials (range 11–25). The first few practice trials were monitored, and any confusion about the technique was clarified. No advice on strategy was given. Participants then completed 3 blocks of 10 trials. Short breaks were allowed between blocks.

Workload questionnaires were completed after each technique. The subjective and background questionnaires were completed after all techniques. Sessions lasted about an hour.

5.3.1 Results

ANOVA was strongly non-significant for mean block completion times for PSG ($F(2,10) = 0.06, p = .94$) and TR ($F(2,10) = 0.55, p = .58$), indicating that performance was similar for all blocks in those conditions.

However, for the PSNG condition, ANOVA showed a significant difference for block ($F(2,10) = 3.79, p = .034$). A post-hoc Tukey HSD test found no difference between blocks at the .05 significance level, but at the .065 significance level found differences between block 1 and block 2 and between block 1 and block 3 . Block means over all participants were 17.3 s, 12.9 s, and 13.1 s. The greater mean for block 1 is likely due to the lack of practice time for PSNG—participants needed time to habituate to the absence of the graphical display, but their performance stabilized in blocks 2 and 3. For this reason, only data from blocks 2 and 3 of the PSNG condition was used in the analysis below.

Trial time data for one participant was improperly recorded and had to be discarded.

Quantile-normal plots of trial times revealed that the time data was non-normal, but log-transformed data was approximately normal. Thus, the log-transformed values are used for the remainder of the analysis.

Performance in TR was about 15% faster than performance in PSG ($t(10) = 2.32, p = .043$), contradicting our first hypothesis.

Performance in PSNG was significantly faster (14%) than in PSG ($t(10) = 3.56, p = .005$), confirming our second hypothesis. However, this difference must be treated with caution, as it confounds the effect of the two techniques and the effect of greater practice before PSNG (since PSG always came before PSNG). Note again that the comparison does not include block 1 of PSNG, which we treated as a practice block and which had higher

Table 5.1: Glance counts for all participants in Study 3 for the PSG condition. Rows represent blocks in descending order. The data suggest two groups of users: those who glanced frequently throughout the session, and those who learned to hardly glance at all. Participants are arranged according to group in the table. (From [61]. Copyright 2006 ACM Press. Reprinted with permission.)

Frequent								Seldom			
1	2	3	4	7	8	11	12	5	6	9	10
21	53	64	58	60	59	48	48	7	41	3	35
18	38	57	58	51	54	45	74	0	31	0	20
25	33	49	61	50	54	46	61	0	7	1	3

trial times than blocks 2 and 3.

No significant difference in subjective workload was found between either PSG and PSNG ($t(11) = 1.63, p = .13$) or PSG and TR ($t(11) = 0.16, p = .88$). This confirms our third hypothesis but contradicts our fourth.

Examination of screen recordings revealed that participants glanced at the controls frequently for the TR condition, as expected. I felt that quantitative analysis was not necessary to confirm this result. Glance counts per block for each participant for the PSG condition are given in table 5.1. These data confirm our observation in Study 2 that there are two distinct groups of participants: those who glance for nearly every parameter modification, and those who learn to hardly ever glance. Examination of the screen recordings of participant 1, who glanced throughout but had markedly lower glance counts than the other frequent glancers, revealed that they usually glanced only at the bold/italic/underline controls.

In subjective questionnaires, many participants identified the same three issues with Pokespace's design as in Study 2. Nonetheless, 7 out of 12 participants preferred Pokespace to the traditional technique, with some describing it as “natural”, “smooth”, “convenient”, and “almost instinctive”.

5.4 Discussion

The behaviour of participants with respect to reliance on Pokespace's visual display was surprising and puzzling. Most participants consistently glanced at the display throughout their sessions, yet their performance was not adversely affected when the display was removed. This result has implications for two of the design guidelines in section 3.1.2.

First, while it may be argued that Pokespace did promote visual attention on the object of interest, especially in the PSNG condition of Study 3, this appears to have had little effect on performance. This suggests that the mental workload required to glance at a familiar set of controls may not be particularly high, which casts doubt on the utility of Design Guideline 5. However, I do not believe it should be hastily abandoned, since it seems that a reduction in gaze shifts could certainly not be *injurious* to performance, and it remains unclear whether it would have an effect for higher level, more complex, more realistic tasks. Future studies must examine this question.

Second, rehearsal as described in Guideline 1 requires a *smooth transition* from visual to haptic use. Such a transition was observed for only some users. Additionally, since a reduction in gaze shifts did not appear to improve performance, it is not clear that haptic use is indeed an "expert use". This may even explain why some participants did not voluntarily make the transition to haptic use. Whether haptic use can be considered an expert use, and perhaps also whether a smooth transition from visual to haptic use can occur, depends on whether a reduction in gaze shifts is shown to improve performance in the future. Nonetheless, practice with Pokespace did produce an important rehearsal effect — although disoriented when graphical feedback was removed, participants had learned the location of the text controls well enough that they quickly accommodated and could locate the controls relying only on haptic cues.

Design Guideline 3, which suggests that the haptic rendering should be simple and consistent in order to support the development of motor memory, seems to have had the intended effect. Participants learned to use Pokespace quickly, and very few instances of participants selecting the wrong boundary segment were observed.

As described in section 5.2.1, three issues with Pokespace's design were identified: trouble with bold/italic/underline controls, pull-away errors, and unexpected parameter changes on pop-in.

The first of these issues is possibly due to poor temporal integration, as described in Haptic/Motor Property 1. In particular, participants likely were not able to cumulatively keep track of the position of their hand with respect to the Phantom's coordinate system, and therefore had difficulty knowing which of the three segments of the lower boundary the Phantom tip was closest to. It was hoped that the centering force described in section 3.3.3 would counteract this limitation by providing a spatial cue, but it appears this was not the case. In future, I plan to experiment with increasing the magnitude of this force, and with adding textures to the square's area to act as an additional spatial cue.

The problem of pull-away errors is likely due to the deviation from Design Guideline 6 as described in section 3.3.4. While pulling away from a boundary segment, users are required to restrict their movement orthogonally to the segment, or risk inadvertently changing the parameter value. This should be corrected in future designs.

Finally, the problem of unexpected parameter changes on pop-in is due to the absolute mapping of boundary segment position to parameter value. In future, I plan to experiment with a relative mapping instead. Not only would doing so address this problem, it would also allow the use of more dynamic, velocity based techniques, such as those used by Oakley et al. [48], to reduce unwanted sensitivity in the interface, thus potentially reducing pull-away errors as well.

While it is not clear exactly why the standard GUI controls outperformed Pokespace in Study 3, it is likely that the three above issues contributed to the phenomenon. As measures are taken to remedy those problems, I expect that the performance of Pokespace will continue to improve.

Chapter 6

Conclusion and Future Work

I have argued that a new approach to haptic interaction technique design, as set down in a series of design principles that I developed, will lead to more efficient interaction for everyday computing tasks. Pokespace is a novel haptic bimanual interaction technique intended to demonstrate that claim. Unfortunately, the initial studies conducted in this thesis have not shown Pokespace to be faster than a standard interface, and thus no concrete evidence of the above claim has been produced.

Nonetheless, I believe that there is cause to be optimistic about this work, and to continue its development. Firstly, much has been learned about the nature of the haptic bimanual interface. Study 1 has demonstrated the utility of haptic constraints in improving performance, shown bimanual coordination to be a questionable goal, and highlighted the adaptability and upredictability of humans in developing strategies for interaction tasks.

In Studies 2 and 3, Pokespace has been shown to support the performance of a command selection and parameter specification task with little or no visual attention. Pokespace has also demonstrated that novice users can unconsciously rehearse the ability to perform a task using only haptic feedback, even in as little as 30 trials. In addition to these advantages, and despite the newness of the Pokespace design, user performance in Study 2 was only slightly slower than for a standard interface. Studies 2 and 3 have also suggested several improvements to Pokespace, and to the design principles underlying it.

It is difficult at this point to make inferences as to the fitness of those design principles, since this thesis has not tested them directly, and Pokespace, the interface constructed according to them, has yet to exhibit superior performance. Nonetheless, some of them have been partially validated, in that when they were ignored, problems resulted. Of the three design problems identified with Pokespace, one was likely due to a lack of support of integrality of movement (Guideline 6), and another was seemingly due to poor human temporal integration abilities (Haptic/Motor Property 1). Additionally, Guideline 1 ('design for rehearsal') seems to have had the desired effect in Study 2, while Study 1 provided evidence in support of Guideline 4 ('maximize the number of backstops').

With these lessons and encouraging results in hand, I plan to continue to improve upon Pokespace and the set of design principles, and to subject them to further evaluation. Mitigation of the problems identified in Studies 2 and 3 will be my first concern, and I have two main ideas for that purpose. First, I plan to change the design of the force field in each tool's plane, abandoning the current boundary square layout, since the empty space in the centre of the square has caused problems. Second, I plan to switch to a relative segment-to-value mapping, rather than the current absolute mapping, in order to eliminate unwanted sensitivity. I expect that these two measures will greatly increase the fidelity of the system, without sacrificing its fundamental principles.

I also plan to experiment with other haptic features to increase the richness of the haptic experience of Pokespace. For example, work by MacLean and Enriquez [43] on *hapticons* suggests that users can readily learn mappings of vibrotactile impluses of varying frequency, amplitude, and waveform to arbitrary entities. I expect that adding hapticons to Pokespace may increase the distinguishability of tools and parameters, allowing more fluent eyes-free interaction.

If these enhancements yield positive results, further comparisons of Pokespace to current techniques would be appropriate. Such studies will involve more realistic tasks and more complex implementations of Pokespace, in order to more closely approximate a real-world interaction scenario.

Pokespace need not be limited to the Phantom device — a novel device specifically tailored to the needs of Pokespace could be cheaper and more ergonomically correct, while also improving performance. In particular, I am interested in using a knob for the adjustment of parameters. I estimate that a knob would allow both easier and more precise fine tuning of parameters, as well as fluent selection across a wider range of possible values.

In addition to the original purpose of developing an effective interaction technique, I feel that research with Pokespace can also serve to investigate more basic questions in areas such as psychology and motor control. Human attention is a phenomenon closely related to Pokespace which could benefit from investigation in the context of human-computer interaction. For example, Study 2 revealed that participants were able to shift their visual attention from the word display to the interface controls rapidly and with little or no effect on their performance; but it remains to be seen whether such frequent shifts of attention would be injurious to performance of a more complex task, and if so, why. Meanwhile, Study 1 revealed that users did not develop bimanual coordination in the experimental task. It is not clear why they did not, and under what conditions they might have.

I am enthusiastic about searching for answers to such questions in the future, since a better systematic understanding of human performance is sure to produce a more efficient and enjoyable human computer interface. As this thesis has argued, I believe that haptic feedback is destined to play a central role in that interface of the future. I look forward to the day when the haptic experience provided by a typical computer is equal to the versatility of the tasks of which it is capable, and when every user has a haptic device on his or her desk. I hope that the work presented in this thesis can serve as one of the first steps of the long march toward that goal.

Bibliography

[1] M. Akamatsu, I. S. MacKenzie, and T. Hasbrouq. A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics*, 38:816–827, 1995.

[2] M. Akamatsu and S. Sato. A multi-modal mouse with tactile and force feedback. *International Journal of Human-Computer Studies*, 40(3):443–453, 1994.

[3] R. S. Avila and L. M. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings of the 7th conference on Visualization*, pages 197–204, Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.

[4] N. L. Bernstein, D. A. Lawrence, and L. Y. Pao. Design of a uniactuated bimanual haptic interface. In *Proceedings of IEEE HAPTICS 2003*, pages 310–317, 2003.

[5] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager. Recognition of operator motions for real-time assistance using virtual fixtures. *IEEE Transactions on Robotics*, 20(6):953–866, 2004.

[6] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: The see-through interface. In *Proceedings of ACM SIGGRAPH 1993*, pages 73–80. ACM Press, 1993.

[7] K. R. Boff and J. E. Lincoln, editors. *Engineering Data Compendium: Human Perception and Performance.* AAMRL, Wright-Patterson AFB, OH, 1988.

[8] F. P. Brooks, M. Ouh-Young, J. J. Batter, and P. J. Kilpatrick. Project GROPE: Haptic displays for scientific visualization. In *Proceedings of ACM SIGGRAPH 1990*, pages 177–186, 1990.

[9] W. Buxton and B. Myers. A study in two-handed input. In *Proceedings of ACM CHI 1986*, pages 321–326, New York, NY, USA, 1986. ACM Press.

[10] F. J. Clark and K. W. Horch. Kinesthesia. In K. R. Boff, L. Kaufman, and J. P. Thomas, editors, *Handbook of Perception and Human Performance*, volume 1, pages 13–1–13–62. Wiley, 1986.

[11] H. Delingette and N. Ayache. Hepatic surgery simulation. *Communications of the ACM*, 48(2):31–36, 2005.

[12] J. T. Dennerlein, D. B. Martin, and C. Hasser. Force-feedback improves performance for steering and combined steering-targeting tasks. In *Proceedings of ACM CHI 2000*, pages 423–429, 2000.

[13] F. L. Engel, P. Goossens, and R. Haakma. Improved efficiency through I-and E-feedback: A trackball with contextual force feedback. *International Journal of Human-Computer Studies*, 41(6):949–974, 1994.

[14] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison Wesley, 1990.

[15] W. R. Garner and 225-241. G. L. Feleoldy 1970. . Cog. Psychol. 1. Integrality of stimulus dimensions in various types of information processing. *Cognitive Psychology*, 1:225–241, 1970.

[16] R. C. Goertz and R. C. Thompson. Electrically controlled manipulator. In *Nucleonics*, pages 46–47, 1954.

[17] J. Grosjean, J. M. Burkhardt, S. Coquillart, and P. Richard. Evaluation of the Command and Control Cube. In *Proceedings of ICMI 2002*, pages 473–478, 2002.

[18] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 9(4):486–517, 1987.

[19] F. Guimbretière, A. Martin, and T. Winograd. Benefits of merging command selection and direct manipulation. *ACM Transactions on Computer-Human Interaction*, 12(3):460–476, 2005.

[20] F. Guimbretière and T. Winograd. Flowmenu: Combining command, text, and parameter entry. In *Proceedings of ACM UIST 2000*, pages 213–216, 2000.

[21] S. G. Hart and L. E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In P. Hancock and N. Meshkati, editors, *Human Mental Workload*, pages 139–183. North Holland B.V., Amsterdam, 1988.

[22] V. Hayward and M. Cruz-Hernandez. Tactile display device using distributed lateral skin stretch. In *Proceedings of IEEE HAPTICS 2000*, pages 1309–1314, 2000.

[23] K. Hinckley, R. Pausch, D. Profitt, J. Patten, and N. Kassell. Cooperative bimanual action. In *Proceedings of ACM CHI 1997*, pages 27–34, 1997.

[24] C. Ho, C. Basdogan, and M. A. Srinivasan. Efficient point-based rendering techniques for haptic display of virtual objects. *Presence*, 8(5):477–491, 1999.

[25] I. Hopkins, J. Callahan, and M. Weiser. Pies: Implementation, evaluation, and application of circular menus. Technical report, Computer Science Department, University of Maryland, 1988.

[26] R. Jacob, L. E. Sibert, D. C. McFarlane, and J. R. Mullen. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction*, 1(1):3–26, 1994.

[27] R. S. Johansson and G. Westling. Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects. *Experimental Brain Research*, 56(3):550–564, 1984.

[28] P. Kabbash, W. Buxton, and A. Sellen. Two-handed input in a compound task. In *Proceedings of ACM CHI 1994*, pages 417–423, New York, NY, USA, 1994. ACM Press.

[29] A. E. Kirkpatrick and S. A. Douglas. Application-based evaluation of haptic interfaces. In *Proceedings of IEEE HAPTICS 2002*, pages 32–39, 2002.

[30] R. L. Klatzky and S. J. Lederman. Touch. In A. F. Healy and R. W. Proctor, editors, *Experimental Psychology*, volume 4, pages 147–176. Wiley, New York, 2002.

[31] R. L Klatzky, J. M. Loomis, S. J. Lederman, H. W. Wake, and N. Fujita. Haptic identification of objects and their depictions. *Perception and Psychophysics*, 54:170–178, 1993.

[32] R. Komerska and C. Ware. Haptic task constraints for 3D interaction. In *Proceedings of IEEE HAPTICS 2003*, pages 270–277, 2003.

[33] R. Komerska and C. Ware. A study of haptic linear and pie menus in a 3D fish tank VR environment. In *Proceedings of IEEE HAPTICS 2004*, pages 224–231, 2004.

[34] G. Kurtenbach and W. Buxton. The limits of expert performance using hierarchic marking menus. In *Proceedings of ACM InterCHI 1993*, pages 482–487, 1993.

[35] G. Kurtenbach and W. Buxton. User learning and performance with marking menus. In *Proceedings of ACM CHI 1994*, pages 258–264, 1994.

[36] G. Kurtenbach, T. P. Moran, and W. Buxton. Contextual animation of gestural commands. *Computer Graphics Forum*, 13(5):305–314, 1994.

[37] S. J. Lederman and R. L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19:342–368, 1987.

[38] S. J. Lederman and R. L. Klatzky. Extracting object properties through haptic exploration. *Acta Psychologica*, 84:29–40, 1993.

[39] A. Leganchuk, S. Zhai, and W. Buxton. Manual and cognitive benefits of two-handed input: An experimental study. *ACM Transactions on Computer-Human Interaction*, 5:326–359, 1998.

[40] R. G. Lipsey, K. I. Carlaw, and C. T. Bekar. *Economic Transformations: General Purpose Technologies and Long-Term Economic Growth.* Oxford University Press, 2005.

[41] J. M. Loomis. Distal attribution and presence. *Presence: Teleoperators and Virtual Environments*, 1:113–119, 1992.

[42] I. S. Mackenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7:91–139, 1992.

[43] K.E. MacLean and M.J. Enriquez. Perceptual design of haptic icons. In *Proceedings of Eurohaptics 2003*, Dublin, Ireland, 2003. 13 pages.

[44] T. H. Massie. Initial haptic explorations with the Phantom: virtual touch through point interaction. Master's thesis, Massachusetts Institute of Technology, 1996. Unpublished.

[45] T. Miller and R. Zeleznik. An insidious haptic invasion: Adding force feedback to the X desktop. In *Proceedings of ACM UIST 1998*, pages 59–64, 1998.

[46] D. Norman. *The Design of Everyday Things.* MIT Press, Cambridge, MA, 1998.

[47] I. Oakley, A. Adams, S. A. Brewster, and P. D. Gray. Guidelines for the design of haptic widgets. In *Proceedings of British HCI 2002.* British Computer Society, 2002. 16 pages.

[48] I. Oakley, S. A. Brewster, and P. D. Gray. Solving multi-target haptic problems in menu interaction. In *Extended Abstracts of ACM CHI 2001*, pages 357–358, 2001.

[49] I. Oakley, M.R. McGee, S.A. Brewster, and P.D Gray. Putting the feel in 'look and feel'. In *Proceedings of ACM CHI 2000*, pages 415–422, 2000.

[50] M.S. O'Modhrain and R.B. Gillespie. The Moose: A haptic user interface for blind persons. In *Proceedings of the 3rd World Wide Web Conference*, pages 37–42, 1997.

[51] M. A. Otaduy and M. C. Lin. Stable and responsive six-degree-of-freedom haptic manipulation using implicit integration. In *Proceedings of IEEE HAPTICS 2005*, pages 247–256, 2005.

[52] S. Park, R. Howe, and D. Torchiana. Virtual fixtures for robotic cardiac surgery. In *4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1419–1420. Springer Verlag, 2001.

[53] S. Pook, E. Lecolinet G. Vaysseix, and E. Barillot. Control menu: Execution and control in a single interactor. In *Extended abstracts of ACM CHI 2000*, pages 263–264, 2000.

[54] C. Ramstein and V. Hayward. The pantograph: A large workspace haptic device for a multi-modal human-computer interaction. In *Proceedings of ACM CHI 1994*, pages 57–58, 1994.

[55] J. Raskin. *The Humane Interface: New Directions for Designing Interactive Systems.* Addison Wesley, Reading, MA, 2000.

[56] G. Robles-De-La-Torre and V. Hayward. Force can overcome object geometry in the perception of shape through active touch. *Nature*, 412:445–448, 2001.

[57] L. B. Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 76–82, 1993.

[58] R. Schmidt and T. D. Lee. *Motor Control and Learning*, pages 64–67. Human Kinetics, Champaign, IL, USA, 3rd edition, 1999.

[59] K.B. Shimoga. A survey of perceptual feedback issues in dexterous telemanipulation: Part 2. finger touch feedback. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 271–279, 1993.

[60] B. Shneiderman. Direct manipulation: a step beyond programming languages. *IEEE Computer*, 16:59–69, 1983.

[61] T. N. Smyth and A. E. Kirkpatrick. A new approach to haptic augmentation of the GUI. In *Proceedings of ACM ICMI 2006*, 2006. 8 pages.

[62] S. S. Snibbe, K. E. MacLean, R. Shaw, J. Roderick, W. L. Verplank, and M. Scheeff. Haptic techniques for media control. In *Proceedings of ACM UIST 2001*, pages 199–208, New York, NY, USA, 2001.

[63] I. R. Summers and C. M. Chanter. A broadband tactile array on the fingertip. *Journal of the Acoustical Society of America*, 112:2118–2126, 2002.

[64] C. Swindells, K. E. MacLean, K. S. Booth, and M. Meitner. A case-study of affect measurement tools for physical user interface design. In *Proceedings of Graphics Interface 2006*, 2006.

[65] A. Tang, P. McLachlan, K. Lowe, C. R. Saka, and K. MacLean. Perceiving ordinal data haptically under workload. In *Proceedings of ACM ICMI 2005*, pages 317–324, 2005.

[66] N. Walker, D. E. Meyer, and J. B. Smelcer. Spatial and temporal characteristics of rapid cursor-positioning movements with electromechanical mice in human-computer interaction. *Human Factors*, 35(3):431–458, 1993.

[67] P. S. Wellman, W J. Peine, G. E. Favalora, and R. D. Howe. Mechanical design and control of a high-bandwidth shape memory alloy tactile display. In *Proceedings of 1997 International Symposium on Experimental Robotics*, 1997. 12 pages.

[68] C. B. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems 1994*, pages 146–150, 1994.