Whole-hand Input

by
David Joel Sturman
M.S., Rutgers University (1981)
S.B., Massachusetts Institute of Technology (1979)

Submitted to the Media Arts and Sciences Section, School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
Massachusetts Institute of Technology
February 1992

Author________________________________
David J. Sturman
Media Arts and Sciences Section
December 20, 1991

Certified by______________________________
David Zeltzer
Associate Professor of Computer Graphics
Media Arts and Sciences Section

Accepted by______________________________
Stephen A. Benton
Chairperson, Departmental Committee on Graduate Students

PUT THE ABSTRACT PAGE HERE

Thesis Committee

Chairman______________________________

David Zeltzer, Ph.D.
Associate Professor of Computer Graphics
MIT Media Arts and Sciences Section

Member______________________________

J. Kenneth Salisbury, Jr., Ph.D.
Principal Research Scientist
MIT Artificial Intelligence Laboratory

Member______________________________

Beth A. Marcus, Ph.D.
President
Exos, Inc.

Member______________________________

Joseph Rosen, M.D.
Associate Professor in Plastic and Reconstructive Surgery
Dartmouth Medical School

January, 1992

to sue,

who carries my heart

Acknowledgements

This has been a lot of fun. Mostly because of all the great people that have been around me, encouraging me on, sharing their work, their ideas, and their friendship. The support has been present around the globe, from as far as Japan, Australia, and New Zealand, and from as near as my own home. To all who have and do take a part in this, thank you!

Specific individuals and organizations stand out in their contribution to this work.

First of all to the ones I've spent the long hours with, thinking, talking, hacking, playing, laughing. I couldn't have been blessed with a better group of compatriots than the Snakepit crew. My work sits on the software foundations and hard work provided by Steve Pieper and Dave Chen, my fellow graduating Ph.D.'s. Next in the Ph.D. line are Mike McKenna, Steve Drucker, Tinsley Galyean, and Michael Johnson, the best support group anyone has ever had. They gave valuable feedback on many details, and were always willing guinea pigs. Margaret Minsky always says the right thing and sheds new light on all my theoretical and practical concerns. I can always count on Peter Schröder for sound advice, critical thinking, and a good dinner wine. Paul Dworkin supplied useful tools, and invented and sustained the ever nourishing Media Lab Friday School Lunch Program.

I thank my principal advisor David Zeltzer for supporting me in this program, providing an environment free from worries of funding or equipment, and the freedom to pursue my own goals even though they did not always match his own. I am grateful for his suggestions and critical eye steering me successfully through this process.

I've been privileged to have Beth Marcus, Ken Salisbury, and Joe Rosen on my committee. All three have an incredible enthusiasm in the work they do, and brought this to me whenever we met. They each contributed in a unique way, together keeping me on course in what was often murky waters.

Tod Machover also contributed much enthusiasm and advice to my work, and I am thankful for the opportunity to have worked with him, for his generosity and patience with errent hardware during the *Bug-Mudra* project, and for a great trip to Japan.

Janette Noss provided invaluable administrative assistance, getting crucial FedEx packages out, procuring equipment and software, handling travel expenses, and generally easing my dealings with the harsh realities of the physical world and its bureaucracies.

Linda Peterson compassionately shepherded myself and other Media Lab graduate students through the labyrinth of class registration, deadlines, and paperwork. I could always count on her to keep me in good graces with the powers that be.

Thanks to Nicholas Negroponte and Jerome Wiesner for the vision and support of the Media Lab.

I am obliged to my sponsors at DARPA, Hewlett Packard, and NHK. Especially to Hideichi Tamegaya and Fumiaki Masuno for their personal warmth and generosity as they supported our work through NHK.

Thank you to everyone at VPL, Exos, Ascension, AGE, and Spatial Systems for technical support and for generously maintaining their devices, which we treated with long, hard overuse and abuse.

I am indebted to fellow colleagues and researchers who have freely shown me their work, shared their ideas, and provided helpful insights and encouragement: Myron Krueger, Jim Kramer, Jock Mackinlay, Gary Grimes, Graham Walters, Scott Fullam, Judy Sachter, Cliff Beshers, and many others.

Bill Henneman helped me find my way in the world of statistics.

I am grateful for the patience of the subjects of my experiments, you shall remain nameless.

Thanks to Bob Sabiston for Figure 4.

I am thankful to my in-laws Bill and Terry for their quiet patience as I all but neglected their daughter when the work got deep. And again to Bill for his editing expertise on the near final draft of this long document.

To my parents, Marianne and Jerry for providing such excellent role models and never letting me forget that I am loved and supported. To my brother John with whom anything can be more fun than you ever thought possible.

Finally, my wife Sue. You stuck out the long graduate student season, sharing my life, and providing care, love, and direction when it was needed the most. It is often hard to see the accomplishment when you have been sustaining the process so hard and for so long. You've done it! It's here! Thank you.

David Sturman
January, 1992

The following trademarks appear in this dissertation:

The Bird™ is a trademark of Ascension Technologies.
DataGlove™ is a trademark of VPL, Inc.
Dexterous HandMaster™ and **DHM**™ are trademarks of Exos, Inc.
Polhemus™ is a trademark of Polhemus
Power Glove™ is a trademark of Mattel
Spaceball™ is a trademark of Spatial Systems
*Talking***Glove**™ is a trademark of *Virtex* Virtual Technologies.
Virtuality™ and **Space Glove**™ are trademarks of W. Industries.

Contents

Contents

List of Figures

List of figures

1 Introduction

In recent years, computer processing speeds have advanced to the point that computers can interactively assist humans in controlling complex tasks. These tasks require simultaneous control of many degrees of freedom. In the cases of manual control, there arise the problem of how to present a control system to the user that affords dexterity, precision, and usability. Conventional input devices, such as dials, mice, and joysticks, with only one, two, or three degrees of freedom, are often ill-suited for these tasks, falling short on one or more of the task requirements. In an effort to address this problem, researchers are developing devices that take advantage of all the degrees of freedom of the hand. Some of these are commercially available and are gaining widespread use. The most prevalent are glove-like, and worn on the hand. Previous and ongoing research with these devices has focused on specific areas such as master-slave robot controllers, "point, reach, and grab" interaction with three-dimensional computer simulations, and sign-language interpreters for the hearing impaired. These models of use address the need for natural user interfaces, but take advantage of only a subset of the capabilities of the hand. There has been little work to examine the full potential of the hand as an input device.

This dissertation examines the full and direct use of the hand's capabilities for the control of computer-mediated tasks. It presents this subject, termed *whole-hand input*, as a distinct study, independent of specific application or interface device.

Whole-hand input derives its usefulness by specifically taking advantage of the hand's qualities of naturalness, adaptability, and dexterity; and that without these qualities, whole-hand input provides little, if any, advantage over conventional interfaces. When these capabilities of the hand are exploited by appropriate use of whole-hand input, many diverse applications can benefit including telerobotics, remote vehicle and equipment control, puppetry, musical performance, surgery, and scientific visualization.

There is more available with whole-hand input than simply "putting" one's hand into a computer simulation. The hand can be used as a sophisticated computer input and control device, managing complex coordinations of many degrees of freedom.

This dissertation takes a disciplined approach to investigating the potential of whole-hand input and its appropriate use for the control of complex task domains. It develops a common basis for the description, design, and evaluation of whole-hand input—an important element for the advancement of the field. This is embodied by the *design method for whole-hand input*, a series of procedures with an accompanying taxonomy that enumerate key issues and points for consideration in the development of whole-hand input. The design method helps designers focus on task requirements, isolate problem areas, and choose appropriate whole-hand input strategies for their specified tasks.

Whole-hand input is a newly emerging study. This dissertation is the first work that comprehensively treats the subject independently of specific application. Although it presents specific techniques and ideas, they are meant as starting points for further study and exploration. A full section of this document is devoted to detailed recommendations for future work. The intent of this dissertation is not to present solutions to previously asked questions, but to underscore the questions and to provide tools and a conceptual framework for future researchers and developers to pursue their own designs.

Reader's guide

Section 2, following this introduction, presents a working definition of whole-hand input and the three salient features of whole-hand input: naturalness, adaptability, and dexterity.

Section 3 discusses how whole-hand input could be used in six application areas: construction, robotics, puppetry and animation, music, surgery, and scientific visualization.

Section 4 covers the issues involved in the development of practical applications of whole-hand input, and describes how the dissertation addresses them.

Section 5 discusses previous work that constitutes the background and context for this dissertation. The section is broken into two parts. The first deals with the general issues of the human-computer interaction. The second deals with specific applications of whole-hand input. This section also describes the large variety of devices that have been used for computer input of whole-hand motion since the master-slave controllers of the 1950's and 1960's.

Section 6 presents the design method for whole-hand input. It is divided into five parts: a test for appropriateness of use, a taxonomy of use, a task vs. hand-action evaluation guide, device capabilities, and a procedure for the use of the design method.

Section 7 describes three experiments that were performed to validate the design method. These experiments were chosen to illustrate the range of contrast between whole-hand input and conventional input.

Sections 8 and 9 contain the description of the testbed for experimenting with whole-hand input techniques and implementation details for the whole-hand input abstraction library.

Section 10 describes three prototype applications, illustrating a variety of whole-hand input techniques. User responses to working with the prototypes are reported with each description. This section also describes the development of the use of whole-hand input for a series of musical performances.

Finally, Section 11 recommends further tests and avenues of research to forward the understanding of whole-hand input.

2 Whole-hand Input

2.1 Definition

Whole-hand input is the full and direct use of the hand's capabilities for the control of computer-mediated tasks. At a functional level, it is the information a computer derives from the monitoring of the individual degrees of freedom of the hand. In the fullest sense of the term, this input involves the 29 degrees of freedom of the hand: 23 degrees of freedom in the hand joints above the wrist (Figures 1, and 2), and 6 degrees of freedom in the free motion of the palm (derived from the wrist, forearm, elbow, and shoulder joint motions). Forces generated by the hand should also be considered in the full description of whole-hand input, however are only briefly treated in this dissertation. At the other end of the spectrum, whole-hand input may be as simple as monitoring the three-space position and orientation of the palm or the bends of three or four fingers. The distinguishing characteristic of whole-hand input is that the user does not think in terms of manipulating an input device, as is the case with other haptic forms of input (e.g., mouse, joystick, trackball), but moves his hand to directly affect the task. A functional way to describe the distinction is that whole-hand input is derived from direct measurement of hand motion rather than measurement of the motion of a device manipulated by the hand.

Some examples of whole-hand input (discussed in more detail in Section 3) are using hand signs to control a teleoperated crane at a construction site; miming the motions of reaching and grabbing to pick and move objects in a computer-simulated scene; flexing different fingers to move the head, arms, and legs of a computer-animated character; tracing space curves with a finger to indicate surgical cuts in a simulated operation; and flexing fingers and moving the hand to modify audio parameters that control the color and tone of a live musical performance.

figs/hand-joints.eps

Figure 1: *The hand joints* There are 17 active joints in the hand, together providing 23 degrees of freedom. The third degree of freedom of the trapeziometacarpal joint (base of thumb) allows the thumb to rotate longitudinally as it is brought into opposition with the fingers (see Figure 2). This rotation is dependent on the other degrees of freedom of the thumb. Thus, it might be said that it is not a true degree of freedom and that the hand joints only embody 22 degrees of freedom. (Diagram adapted from Napier, 1980, p. 29.)

figs/motions.eps

Figure 2: *Motion of the hand joints* Terms for the motions of the fingers are fairly well established. However, the motions of the trapeziometacarpal joint (base of thumb) are subject to a variety of names (Tubiana, 1981). The ones shown here are the most common. (Diagram adapted from the American Society for Surgery of the Hand, 1978.)

Definition 2.1

figs/tendons.eps

Figure 3: *Muscles and tendons of the back of the hand* The many interconnections and interactions of the muscles and tendons of the hand give rise to the complexity of hand motion. Most of the muscle mass lies in the forearm with long tendons transmitting power to the fingers. This arrangement allows the hand to be light and flexible without sacrificing strength. (Diagram adapted from *Hand and wrist*, a wall chart by the Anatomical Chart Co., Stokoe, IL, 1988.)

2.2 Features of whole-hand input

The salient features of using the whole hand as an input device can be divided into three principal categories: naturalness, adaptability, and dexterity.

Naturalness is used to describe a subjective evaluation of interaction. It implies being "free from artificiality, affectation, or constraint," and "obviously suitable for a specific purpose" (Webster's Dictionary, 7th edition).[1] Naturalness in whole-hand input is a function of the hand being used every day for a broad spectrum of tasks using a repertoire of skills that require little thought. These skills can be called *pre-acquired sensorimotor routines*, or *pre-acquired skills*. Examples include different types of grips, specific finger coordinations, such as those used to turn objects within the grasp, and rhythms used in finger tapping and other repetitive motions. It is possible that tasks can be made easier to learn and master by taking advantage of pre-acquired skills in whole-hand input, reducing training expense and time.

Actions that are natural also tend to be intuitive or ingrained behaviors. This can be advantageous, as in an emergency situation where an instinctual reaction, such as opening the hand to release, produces the correct result; or dangerous, as with the early USAF F-111 swing-wing aircraft in which the faster wing configuration was initiated by pulling the control stick backwards—a motion that means "slower" to pilots—leading to several landing accidents before the controls were reversed (Sexton, 1988). On occasion, interface designers can take advantage of natural or ingrained behaviors, but otherwise must be mindful of not conflicting with natural or trained responses.

Another aspect of naturalness has to do with using the hand to sign and signal. Workers already use hand signals to communicate, e.g., in aircraft taxiing, vehicle docking, and crane operation. Different cultures have established different sets of hand signs for simple instructions such as "come here" or "stop" (Morris, 1977). These same signs can be used for computer input, relying on already established and practiced lexicons. Again, using familiar actions would reduce the learning curve and improve operator performance.

[1]Although naturalness implies free from constraint, there are many situations in which external constraints to the hand are helpful. Section 4.4 discusses this issue. For the most part, however, this dissertation discusses the use of the hand free from external forces or contacts.

A third aspect of naturalness is that whole-hand input tends to be body-centered. When objects are manipulated with whole-hand input, a natural coordinate system centered on the body is used, often making the task easier than it would be with a joystick or other device-centered tool. Manipulating objects in coordinate systems that are not body-centered requires extra cognitive effort to perform the body-space to control-space to task-space mapping. We have observed in our own work that body-centered coordinate systems are more natural to work with in many situations and can improve performance for object manipulation tasks.

Finally, the absence of an intermediary device brings whole-hand input a step closer to the experience of direct manipulation by apparently putting the body in direct contact with the objects of interest. This aspect of using the hand to immediately affect a situation contributes to an increased sense of presence, an important element in the successful development of teleoperated systems (Minsky, 1980; Sheridan, 1989).

Adaptability refers to the hand's ability to quickly and smoothly switch functions. For example, lifting a heavy object into place, carefully aligning it with adjoining supports, and then fitting small screws into small holes to secure it. The hand capability that allows all of this enables us to use the same whole-hand input device for a variety of functions, freely switching between modes of control without having to change program modes. This adaptability can be an advantage in situations where different tasks need to be performed but physical space is too limited to have a different input device for each task, or the device transitions are too slow and cumbersome, interrupting the flow between tasks.

For example, military pilots are faced with this problem. There are innumerable functions they need to perform in a small space under great physical, mental, and temporal constraints. Cockpit designers go to great lengths to place as many controls as possible on the control stick or near the pilot's reach. If the number of switches, buttons, dials, and levers are to be reduced, then the resulting interface must assume a wide variety of functions.

The adaptability of the hand is a result of its structure and variety of muscles. Thirty-nine muscles power the hand and wrist. The placement of heavy *extrinsic* muscles away from the hand in the forearm allows the hand to be flexible and light without sacrificing strength. Smaller *intrinsic* muscles in the hand itself permit independent action of each finger joint (Tubiana, 1981). (Figure 3 shows the muscles and tendons of the back of the hand.) Different muscles have different ranges of function, strength, and precision. Since any part of the hand can be monitored for whole-hand input, computer input controls can be linked with those muscles and motions appropriate for the required task. The ability to dynamically control the impedance of the degrees of freedom of the hand also contributes to the hand's adaptability. Joints can be relaxed or stiffened as needed, manipulating a small flower as easily as a heavy block of wood.

Dexterity can be defined as the integration of movements and senses into higher levels of competence (Salisbury, 1987, p. 353). For example, turning a bolt on a nut is a highly coordinated skill that, once learned, is performed as a single action. The significant aspect of this integration is the ability to draw on known sensorimotor routines and, combining them with practiced sensorimotor control, to manipulate and move the hand in new ways, learning and developing new skills. For example, a Westerner learning to use chopsticks finds the task easier when relying on the already acquired skills for manipulating a pencil. By developing skills, the cognitive load required to accomplish a particular task is reduced. The cognitive and neurological schemes by which this is done are not well understood. Some theorize that through practice of a task, low-level motor programs are developed and stored. In time these can be invoked to run outside the cognitive system (Turvey, Fitch, and Tuller, 1982; Young and Schmidt, 1991).

There are many physical and neurological constraints in the hand that facilitate or enforce useful coordination (e.g., motion of the thumb in opposition—*circumduction*). These biologic effects can be exploited in whole-hand input to reduce the difficulty of complex tasks. One goal of this research is to better understand how to decompose and map complex tasks into "simpler" hand manipulations.

By transforming difficult coordinations into dexterous skills, operators can concentrate on the task rather than the difficulty of the interaction. The interface becomes transparent at that point, improving the operator's sense of presence and connection to the task.

3 Application Areas

The use of teleoperation and computer-assisted manual control is a growing interest in many application domains including underwater, terrestrial, and space robotics; surgical simulation; real-time musical performance; and scientific visualization. In many cases, these applications cannot be performed without human direction or intervention. This often involves the real-time manipulation of many parameters, a difficult task, both in the types of control that need to be imposed on the process and the physical manipulation of the devices that provide the operator that control.

Conventional control devices are inadequate for the simultaneous manipulation of more than a few parameters, especially if coordinated control of many degrees of freedom is involved. They also can limit the dexterity that can be applied to a task, and can inhibit smooth transitions between modes of control. For instance, dials provide only one degree of freedom and are not easily coordinated in sets greater than two (one per hand).[1] Levers exhibit the same limitations. Sliders are a slight improvement in that adjacent sliders can sometimes be controlled with different fingers of the same hand. Mice allow two simultaneous degrees of freedom with the addition of buttons for discrete input, however the muscles controlling the wrist and arm, used for operating a mouse, are not the most precise muscles of the hand-arm system, and thus are not well matched for all mouse operations. A data tablet is better for precision operations, because it allows the use of the fingers in a precision grip to control fine motions, but still limits the user to two degrees of freedom.[2]

In all of these cases the full dexterity of the hand-arm system is underutilized. This is

[1] There are cognitive limits to the number of tasks that a person can manage at once. In some cases, more than two dials are difficult to control at a time only because it is difficult to concentrate on more than two operations at a time. It is possible that the cognitive load can be reduced by integrating controls into higher-level, whole-hand input sensorimotor routines. Graham Walters, of Pacific Data Images, a computer animation production house in Sunnyvale, CA, relates an incident in which a professional puppeteer used both hands on eight dials to simultaneously control the eight degrees of freedom of his computer-controlled puppet (personal communication). Graham was astounded that the performer (puppeteer) was able to get life-like motions out of the puppet in just a few minutes of practice on the dials. It seems that expert puppeteers can manage almost any control structure, given enough time. It is unclear if they do this by integrating the controls into dexterous motions, or if they simply have trained themselves to control many things at once.

[2] There exist three-degree-of-freedom data tablets where pen pressure is measured in addition to x and y, and six-degree-of-freedom data tablets that sense the pressure, orientation, and roll of the pen on the tablet; however the range of orientations is limited (e.g., the tip must point towards the tablet).

not to say that these devices fail; they are very effective for certain tasks. However, when used as generic input devices, they can be inefficient in the many situations where the task requires more or different degrees of freedom than those of the device, or where the task is better suited to a different class of hand shape or muscle control than afforded by the device. Whole-hand input allows all of, or any part of the hand to be used and thus allows input strategies that can appropriately fill the task requirements.

The remainder of this section discusses six application domains and how whole-hand input can improve user interaction for each. These applications each reflect the importance of a different aspect of whole-hand input. Together they cover a wide range of tasks, control requirements, and user cultures. Prototypes of three of these applications were simulated with the whole-hand input testbed: a mobile robot with multiple manipulators, a construction crane, and an interactive computer graphic puppet (Sections 10.1, 10.2, and 10.3 respectively). A fourth application was implemented in the area of musical performance and is reported in Section 10.4.

The basic whole-hand input control loop for all of these applications can be generalized in the following manner (diagrammed in Figure 4):

- The user has some concept of a task to be performed.
- The control for the task is formalized as a group of hand motions or gestures that correspond as closely as possible to the conceptual model of the task, maximizing the "naturalness" of the control motions, taking into account factors such as precision, coordination, steadiness, and adaptability of those hand motions as required by the task.
- As the user executes the hand motions, the computer reads the motions and either passes them through directly (e.g., hand position = robot end-effector position), transforms them into control signals (e.g., thumb and forefinger flex = tightness of gripper), or interprets them as symbols in a lexicon (e.g., quickly closing the fist = "stop").
- Ideally, feedback may take many forms including direct visual observation, graphical presentation or simulation, haptic feedback, and sound.

figs/input-loop.eps

Figure 4: *Whole-hand input control loop* Shows the control loop from conceptual model, to hand motion, to computer interpretation, to task execution. Ideally, feedback can take many forms, including visual, haptic, and acoustic.

3.1 Remotely controlled vehicles and manipulators

There is a great interest and need for exploring and working in hazardous environments, be it under the ocean, inside a nuclear reactor vessel, or in space. The dangers of these environments make it costly or impossible for humans to operate in, and mechanical surrogates must be used. To use robots to perform complex tasks, an intelligent autonomous robot, a high degree of manual control, or some combination of the two is required. In cases of manual control, multiple degrees of freedom must be coordinated. Even though the operator is physically removed, a strong sense of presence with the objects to be manipulated is desired. The effects of being physically removed from the task (such as communications delay, cognitive mapping from hand motion to task motion, and absence of physical feedback) should be minimized (Minsky, 1980; Sheridan, 1989). Telepresence systems can drastically reduce the training necessary for teleoperation (Chandler, 1990, paraphrasing M.I.T. professor Harold Alexander). For this reason, researchers are developing head-mounted displays (Fisher et al., 1986), ergonomic and isomorphic manipulator controls (Robotics World, 1989), and kinesthetic and tactile feedback devices (Iwata, 1990; Minsky et al., 1990).

There are several areas in which whole-hand input can contribute to the solution of the problems of remote control. The first is in handling the problem of coordinating many degrees of freedom at the same time. Consider, for instance, the operation of an undersea manipulator. The main systems requiring control include propulsion, vision (cameras and lights), and manipulators to perform various tasks. These systems may all be under real-time human control or cooperative human-computer control. Cooperative human-computer control may be used in the vision system, for instance, where the operator controls a manipulator arm and the computer keeps the focus of the camera and lights on the tools at the end of the arm.

Conventional interfaces to these systems usually involve an array of switches, dials, buttons, levers, and perhaps a joystick or two.[3] The operator is constrained to controlling only one or two degrees of freedom at a time, and must constantly move his hands back and forth over the control-board surface pulling his attention away from the deep-sea monitors to be certain of reaching for the proper control. If time is a critical factor in the execution of a task, then conventional input devices may not provide the necessary dexterity and control. In some cases conventional devices may actually make the task more difficult than one might expect (McKinnon, King, and Runnings, 1987). An additional factor may be the time and concentration lost in switching from device to device that can be avoided if the conceptual functions of several conventional devices are mapped to different modes of one whole-hand device.

Whole-hand input takes full advantage of the hand's dexterity. Thus, it is an excellent candidate for use in these situations. For instance, the motions of one hand can be mapped directly to the motions of an underwater manipulator, while the other hand directs a high pressure nozzle, cleaning debris off the object being manipulated. Both are six-degree-of-freedom (or greater) problems which can be managed by natural hand motions. A very fast hand motion (e.g., flick of the wrist, or some other decoupled hand motion) could be used as a clutch to disengage or switch modes to other sub-tasks, so that the operator can keep his eyes on the work area (Sturman, Zeltzer, and Pieper, 1989).[4] Section 10.1 describes a prototype application involving the operation of a mobile robot with whole-hand input.

[3] Based on personal observations of systems at the Woods Hole Oceanographic Institute, and communications with researchers in the field.

[4] See the discussion on *appropriate response*, page 49.

3.2 Construction industry

One of the key tasks in the construction industry (for which there is a shortage of skilled labor) is the operation of heavy machinery, such as cranes (Kangari and Halpin, 1989; Skibniewski and Russell, 1988). The difficulty of crane operation is moving a large mass at the end of a long, compliant lever arm, controlled by nonlinear hydraulic actuators, using a single lever per degree of freedom interface (Sutton, Cherrington, and Towill, 1986). In other words, the task is one of controlling a multi-degree-of-freedom system which has nonlinear response. In addition, the crane operator often cannot see, or has a distant view of the load at the end of the crane and relies on hand signals from another laborer at the load end (Figure 5).

Hughes et al. (1989) describe the problems one equipment manufacturer had in developing a control interface for a pipe-lifter. The original design had eight levers each controlling one of eight degrees of freedom. This proved intractable for operators because the linear arrangement of the levers, combined with the nonlinearities of the system resulted in an unintuitive mapping into the control space. Hughes et al. solved the problem with a more intuitive double-joystick interface and processors to linearize the control task. More often now, micro-processors are being used to minimize the nonlinear effects of the actuators (Cosgrove, 1990), but additional work is needed towards developing intuitive user interfaces.

figs/crane-signals.eps

Figure 5: *Hand signals for crane operation* Some of the construction industry hand signals used to communicate to crane operators . (Courtesy of Marr Equipment Corporation, Boston, MA.)

Through interactive graphical simulation, this dissertation work will show how natural whole-hand input techniques can be used for the control of this machinery (Section 10.2. By allowing workers to communicate directly with the equipment using the same hand signals and motions they use now to communicate to the equipment operator, the operator can stand at the load end controlling the crane with natural and familiar hand motions. This idea can be extended to the control of other construction equipment such as back-hoes, loaders, and lifters. Other mappings can be developed linking hand motion to equipment behaviors, improving control and coordination. Examples include direct control methods, where the angle of a finger controls the angle of a degree of freedom of the equipment, and indirect control methods such as using hand signals to direct the positioning of the load. This application is particularly interesting because there is the potential to reduce the training time required to handle this type of machinery, significantly alleviating the problem of the industry's shortage of skilled labor.

3.3 Puppetry and computer animation

In the entertainment industry, remote manipulation is used mainly for puppetry to bring strange and unusual characters to life. The sophisticated puppets used in film production often have many degrees of freedom, and require several puppeteers, each controlling a specific aspect of the character. For each aspect, be it the eyes, cheeks, head position, or other motion, the degrees of freedom are controlled by custom-built devices termed "waldos" (after a character and his inventions in a story by Heinlein, 1942). For example, Slimer, one of the ghosts in *Ghostbusters II*, was controlled by five puppeteers each using a standard joystick (Eisenberg, 1989, p. 21).

The recent push in the animated entertainment industry is towards realism. This implies complexity. The more complex the character, the more sophisticated the control structure needs to be, the more puppeteers are needed to interactively control it, and the more difficult and expensive it is to operate. Better interfaces and control structures will save time and money—critical factors in this industry.

Some work already has been done towards coordinated real-time control of computerized characters, both as computer graphics animations and as physical puppet animations. Ginsberg and Maxwell (1983) and Purcell (1985) used a body suit mounted with LEDs that allowed a computer to monitor a performer's motions in real-time. The performer would act out the part of the character with the computer driving the synthetic character to match the performer's motions. DeGraff and Wahrman developed a generalized controller that they use to give a puppeteer real-time control of a parameterized computer graphic face[5] (Robertson, 1988). It has been used in many contexts, including a live performance at the 1988 SIGGRAPH conference film show and to control the computer-animated face of the persona of the "evil robot" in the 1990 motion picture *RoboCop II*.

Pacific Data Images developed a real-time computer graphic puppet for a weekly television series *The Jim Henson Hour* (Walters, 1989). Using a simple armature, a puppeteer controlled the position, orientation, and jaw movement of a computer generated character appropriately named "Waldo C. Graphic." The puppeteer saw a simplified computer graphic representation of the character superimposed on the live video of the other puppets. Pacific Data Images took the movement data from the original performance and as a post-process, applied it to a more complex representation of the character, adding dynamic features (dangling legs, floppy body, and so forth) in a non-real-time mode. Computer-mediated control of puppets is advancing rapidly and some production companies appear to be developing proprietary general purpose control systems to improve puppeteer capabilities.

Computer animators are confronting similar issues. Complex characters have too many degrees of freedom to easily control. Zeltzer (1985) calls this *the degrees-of-freedom problem*. He classifies computer character animation into three levels: the guiding level, the animator level, and the task level. At the guiding level the animator sets a few parameters, or positions a few degrees of freedom at a time, defining key poses. The computer then interpolates these keys to achieve motion (Girard, 1987; Gómez, 1985; Williams, 1982). Each degree of freedom must be set every time its motion changes (key positions—typically every 3 to 10 frames). A character with fifty degrees of freedom could easily require 200 settings per second of animation (30 frames).

At the animator level, motions can be specified algorithmically. In animator-level systems, animators have control over certain inputs to the algorithm, which then determine the motion (Hanrahan and Sturman, 1985; Reynolds, 1982). Although there is less information to specify in animator-level systems than in guiding-level systems, there is a relinquishing

[5] The face is modeled as a polygonal mesh that can be geometrically distorted in predefined manners based on the setting of several parameters. The control parameters are organized so that they perform coordinated distortions that give the face the appearance of frowning, opening the mouth, closing the eyes, and so on. This is based on original work by Parke (1982).

of fine control to the motion algorithms. Task-level animation is an attempt to handle the degrees-of-freedom problem by abstracting many low-level motions into a few high-level tasks, giving the animator supervisory control over the motion. Examples of task-level control are “walk,” “run over there,” or “pick up that block” (Bruderlin and Calvert, 1989; McKenna and Zeltzer, 1990; Zeltzer, Pieper, and Sturman, 1989). However, this still leaves unresolved the problem of how to put the animator in direct, coordinated control of many input channels.

Whole-hand input can make an important contribution to puppetry in a similar manner to the teleoperation domain. By being able to map more input channels to coordinated hand motions and allowing more natural control schemes, whole-hand input could improve the ability of performers to manipulate complex puppets. In addition, whole-hand input devices could serve as general-purpose controllers that through software can map the motions of the puppeteer (or animator) to motions of a character. This would allow the puppeteer to design the control motions unhampered by the constraints of conventional waldo engineering.

Because the puppeteer can apply the unencumbered dexterity of the hand to the control task, the puppeteer is able to control more complex puppets than with the usual waldo. Using a single general purpose waldo moves the control design problem out of the mechanical engineering domain and into the more adaptable and flexible task of software engineering. Mechanical design is still necessary for the puppet itself. Engineers may also need to provide mechanical supports and restraints on some of the hand’s degrees of freedom to facilitate particular controls and to reduce fatigue—an important component in any control interface.

Section 10.3 describes a prototype implementation of a computer animated puppet controlled by whole-hand input.

3.4 Musical performance

Live musical performance is a demanding activity that requires the simultaneous control of many degrees of freedom with very critical time constraints. The standardization of the MIDI (Musical Instrument Digital Interface) protocol and development of FM synthesis has made computer control and synthesis of the musical process a common practice.

However, until recently, the physical form of electronic instruments has undergone little change. Acoustic musical instruments are designed around specific constraints to produce melodious sound, yet at the same time, allow the musician to exert precise control over many parameters (pitch, tone, attack, duration, volume). Synthesizers remove the acoustic constraints on instrument design and focus attention on interface and performance criteria. As a consequence, artists are beginning to use their whole bodies as input to computer synthesized music (Rokeby, 1988; Trubitt, 1990; Vivid Effects, Inc., 1989).

Whole-hand input in music allows many parameters to be controlled at one time and gives the musician the freedom to move expressively, transmitting that expression to the music. At the M.I.T. Media Lab, Tod Machover has been exploring alternative technologies to control the musical process through MIDI input. His "hyperinstrument" project has been extended to whole-hand input, exploring the use of the hand and fingers to directly control MIDI parameters for synthesized music (Gialanze, 1989). One of the offshoots of this work was his 1989 musical composition *Bug-Mudra*, in which the conductor uses whole-hand input to control the acoustics (and thus color) of the performance in real-time via a MIDI controlled audio mixing panel.[6] A live recording of a 1990 *bug-mudra* performance in Tokyo has been released on compact disk (Machover, 1990).

Musical performance has a rich multi-degree-of-freedom control space that is very different from other applications mentioned here. The control task is highly non-anthropomorphic, the feedback entirely acoustic with no intrinsic physical analogies, and control is time-critical. For this reason it is an excellent domain in which to explore the whole-hand control of abstract, real-time, multi-degree-of-freedom processes.

3.5 Surgical simulation and assistance

Surgeons are beginning to use computer graphics to simulate surgical procedures for training, visual assistance in diagnosis, and prediction of surgical results (Delp and Delp, 1989; Delp et al., 1990; Pieper, 1992). These simulations need to be as realistic as possible. For the simulation to be useful to the surgeon as a training tool or surgical assistant, the surgeon needs to be able to manipulate the graphical representation as if it were the real object.

[6]Section 10.4 describes the application in detail.

As an anthropomorphic form of interaction, whole-hand input has a great advantage over more conventional input devices, and can allow a surgeon to manually interact with a simulation and with simulated tools exactly as with real patients and real tools. A keyboard or mouse can only provide a feeling of being "once-removed" from the task. When patients' lives depend upon a surgeon's manual skills, the distinction is important. In addition, surgeons are reluctant to use unfamiliar tools or procedures unless the learning process is short, or the benefits are enormous. New tools and procedures can use the flexibility of whole-hand input to mimic the methods with which surgeons are most comfortable, improving their acceptance in the operating room.

A specific application of whole-hand input to surgical procedures is in the manipulation of endoscopic tools inside the patient. Current endoscopic tools are conceptually simple devices, usually in the form of a clamp or scissors, having an action of only one degree of freedom. One problem in developing more sophisticated tools is engineering complex devices to fit on the end of the long thin probes. However, a more important difficulty may be controlling more than one or two degrees of freedom at the end of the probe. Because whole-hand input permits a high degree of coordination of many degrees of freedom, controlling endoscopic surgical tools with whole-hand computer input can be a successful strategy towards more sophisticated endoscopic procedures. The success of this strategy could be even more evident in time-critical manipulations, where the uncoordinated control inherent in conventional computer input devices slows down the manipulation process.

Looking further into the future, surgeons will perform precision surgery with remote manipulators inside the body. Whole-hand input does not have the physical constraints present in many of today's surgical instruments. This means that the surgeon can rely on his trained dexterity to perform the fine manipulations necessary without having the constraint of holding and supporting a physical tool.

3.6 Scientific simulation and visualization

Scientists developing computer models of complex natural phenomena need to be able to "steer" their simulations and "visualize" their data. Steering is a process of modifying boundary conditions, solution paths, and state variables to influence the process of a simulation while that simulation is running. Currently, many complex simulations take so long that steering is not an interactive process. However, as computers get faster, many applications that now take hours or days will become real-time interactive simulations. Visualization refers to the real-time display and manipulation of the data of the completed simulation. Both of these situations require the coordinated manipulation of multi-axes data spaces.[7]

Personal discussions with computer scientists helping physical scientists with computer visualization problems has indicated that researchers have a desire to have as direct contact with the simulation as possible.[8] For example, aeronautic researchers want to be able to put their hands (and head) into a simulated fluid flow, shaping its boundary; or grab the flow being simulated and turn it around, peering into the nooks and crannies (Levit and Bryson, 1991).

Not only is this an attempt to get a "feel" for the processes that are going on, it is an attempt to get beyond the barrier many of them feel that conventional input devices present to getting their hands on the data. Scientists are similar to surgeons in this respect. The computer is a tool that is at its best when it is transparent to the application; that is, the computer interface is best when it quickly fades in the user's consciousness and the user experiences working directly with the application task.

Again, like the surgeon, the scientist will benefit from whole hand input since it allows natural interaction with objects—grabbing, turning, pushing, pointing—and a high degree of coordinated control of multi-degrees-of-freedom spaces.

[7] For a review of the some of these issues in the field of scientific visualization see (McCormick, DeFanti, and Brown, 1987).

[8] Personal communications, members of the Ohio Supercomputer Center, the Utah Supercomputing Institute, and NCSA.

4 Issues in Whole-hand Input

There are many issues and problems that need to be resolved before whole-hand input becomes a generally useful tool. The most important of these have to do with distinguishing between appropriate and inappropriate uses of whole-hand input. Studying this problem will yield a better understanding of where, when, and how whole-hand input will improve real-time control.

Determining the appropriate use of whole-hand control for an application should be based on analysis of the application, the tasks to be performed, and the qualities of the control models under consideration. One would expect there to be a set of rational criteria based on theoretical and empirical results, however, the "human equation" is difficult to quantify. Most attempts at user interface theory in general have been difficult to apply in practice, and empirical findings have been resistant to generalization and theory (Carroll and Campbell, 1988; Carroll, 1988). Good interfaces are developed in cycles of design, testing, and redesign, loosely based on previous practice and theory. Nevertheless, to have some basis from which to be able to describe, discuss, contrast, and evaluate different tasks and interfaces, there has been a strong effort in the classification of user interface devices and interface methods (Buxton, 1990; Card, Mackinlay, and Robertson, 1990), and in the development of task assessment (Barnes, 1987; Robinette, Ervin, and Zehner, 1987). Given these two sides of the problem, the need for common bases of description and evaluation, and the need for iterative design, this dissertation concentrates on the organization of ideas and tools for the description, evaluation, and design of whole-hand interfaces.

The purpose of this section is to present the issues of whole-hand input that form the context for and drive the need for the contributions of this dissertation: tools for the development and evaluation of whole-hand input. Not all of the questions raised in this section are answered in this dissertation, however it is hoped that the ideas and tools developed here permit future research to achieve this goal.

The first three issues discussed in this section are the most important. They lay the groundwork for the use of whole-hand input. The first is the issue of *appropriate use* of whole-hand input: when and why should or should not whole-hand input be used in an application. The second is the issue of *appropriate control design*: what is the best way to use whole-hand input with a particular application. The third is the issue of *appropriate device*: what the best device or method is for capturing hand data, given an application's needs.

This dissertation addresses these issues directly with the *design method for whole-hand input* in Section 6. The design method embodies a disciplined approach to ascertaining appropriate use of whole-hand input, a taxonomy for the classification of hand action and task response, and metrics for evaluating whole-hand input techniques and device technologies.

The first half of this section discusses the three primary problems of whole-hand input. The second half discusses other issues of whole-hand input that can be characterized as more focused problems within the unresolved larger context. These are the importance of constraints on degrees of freedom, sensory feedback, the use of gestural language, "point, reach, and grab," and one-handed versus two-handed input.

This dissertation focuses on the three major issues that are the most consequential to future progress. However, all of these problems must be solved, and research into the other specific problems is recommended as important future work.

4.1 Appropriate use

Whole-hand input is a novel interface model that has not previously received careful study. Using the hand to manipulate objects and processes is a natural human behavior and so it seems a logical choice for many computer-based tasks. However, not all tasks may be appropriate for whole-hand input. Certainly applications that involve controlling anthropomorphic manipulators, or grabbing, moving, and turning objects (real or graphical)—functions people perform well with their hands—are good choices, but other applications might do better with alternate forms of input. For example, managing a window-based user interface is well suited to mouse and keyboard input because of the application's textual nature and inherent two-dimensionality (or "2.5-dimensionality"); whole-hand input with its extra degrees of freedom may only complicate matters.

Classification efforts similar to those mentioned at the beginning of this section are needed for whole-hand input. The results must support the classification and discussion of hand positions and motions (*hand actions*), and guide the evaluation of hand actions as input controls to specific applications. Ideally, a well designed classification and evaluation scheme will do not only this, but indicate and reveal novel methods of whole-hand interaction as well. The *design method for whole-hand input* is a step in this direction.

Classification of whole-hand actions Many of the current user input classification schemes start with the degrees of freedom of the input device or abstract input device[1] (Buxton, 1990). With whole-hand input, the hand is essentially the device. The problem is how to usefully describe and quantify the approximately 27 degrees of freedom of the space of the hand, given that these degrees of freedom have assorted grades of interdependency, range of motion, and dynamic capabilities.

Previous attempts to classify the hand can be divided into three broad areas; muscle and joint (physiological) oriented, shape (or symbolic) oriented, and task (or function) oriented. Muscle and joint oriented descriptions are mostly found in the medical literature and are used to describe the physiology of the hand for clinical diagnosis and correction of injury and deformity. In this literature the hand is divided along lines of sensory and muscle innervation, and functional muscle, tendon, and skeletal groups (American Society for Surgery of the Hand, 1978; Tubiana, 1981). Shape oriented descriptions are found in the *mudras* of classical Indian dance, and descriptions of sign languages and finger spelling (Klima and Bellugi, 1979; Stokoe, 1960; Waldron and Simon, 1989). The stress there is on the formation of hand shape and motion. No regard is given to the difficulty of the action or the functional capability of the hand within that action. Some attention is paid to the relation of different hand actions to each other and to transitions between hand actions. Task or function oriented descriptions are used mostly in occupational therapy and robotics studies. They tend to classify hand actions according to their manipulative capability in conventional industrial applications, often using grips as the basis for classification (Cutkosky and Wright, 1986).

A unique aspect of whole-hand input is the separation between hand action and task function, so that although the resulting task output is functional, the hand input can be organized along physiological, symbolic, or functional lines. Different applications and modes of whole-hand input may be best described by one, or a combination of these classifications. The medical field with its physiological classification, the language field with

[1] Abstract devices (also called *virtual devices*) are models of input based on logical rather than physical characteristics. For example, a *locator* device locates a position in two- or three-dimensional space, and can be physically implemented with a variety of physical devices such as a mouse, trackball, tablet, or light pen.

its symbolic classification, and the rehabilitative and robotics fields with their functional classifications, address the problem within the framework of their individual needs. The novel framework of whole-hand input requires a synthesis and evolution of these into a more comprehensive system. This is one of the goals of this dissertation.

Evaluation of whole-hand actions Given a useful classification scheme, it should be possible to describe a wide variety of hand actions that can be used for whole-hand input. However, a method of evaluation is required to determine the usefulness of these hand actions. This can be done by experimentation with the hand action in the application, or by analysis of the hand action based on knowledge of the hand, hand action, and the application.

Experimentation will indicate whether or not the chosen hand action is effective, or if it is better than another hand action that also has been tested. It will not indicate if the hand action is the best choice for the application. A priori analysis is important to be able to pick appropriate candidates from the nearly infinite choice of hand actions. Chosen candidates then can be subjected to experimental evaluation. Unfortunately, the literature does little to analyze the capability to move the hand apart from natural functions, an important attribute of whole-hand input, where hand motion is disjoined from function.

The literature concentrates on low-level physiological capability, task performance, or the cognitive basis of hand function. In a review of hand function assessment techniques, Jones (1989b) divides conventional studies into three areas: muscle and joint function and dysfunction, tactile sensibility, and functional or task-oriented capability. Most studies look at one of these at a time. Muscle and joint function is primarily studied in terms of strength, innervation, and range of motion. Tactile sensibility usually tests threshold nerve response and object recognition. Functional capability usually tests the ability to perform fine manipulations, such as handling small objects, or particular motor functions, such as turning dials (Robinette, Ervin, and Zehner, 1987). Functional tests typically measure the speed at which tasks can be accomplished but not how the use of the hand affects task performance.

The cognitive science and neurophysiology literature examines hand function and movement coordination with the aim of explaining the mechanisms of specific hand functions and behaviors (such as pinching, or preshaping of the hand) (Cole and Abbs, 1986; Kelso, 1982; Lederman and Klatzky, 1987). Although the literature does provide explanations, it does little to help evaluate or predict useful hand motions for a general set of applications.

To be able to analyze and evaluate whole-hand input methods, more must be known about what actions can and cannot be done with hands; not just *why* ability exists, but *what* ability exists. For example, it is important to know the overall ability of the hand to motion, gesture, and control it's own shape; what makes a hand action difficult or easy; and what are the temporal and spatial limits of hand performance, organized along single degrees of freedom and in combinations of degrees of freedom.

More must be understood about such factors as interfinger coordination, cross-coupling of the degrees of freedom, resolution of joint motion control, speed of joint control (especially in repetition), and endurance (Durlach, 1989). These factors have been studied in the literature at the task level, but not at the hand-motion level. For instance, unanswered questions include how accurately the average person can control the angle of the index PIP joint, or how often the ring PIP can be flexed without fatigue, or what the exact correlation is between the motions of the thumb IP joint and the index DIP joint which are interconnected at the muscular level. Addressing all of these is beyond the scope of this dissertation. However, Section 11 suggests a series of experiments that lead in these directions.

One of the problems in evaluating whole-hand input techniques is the discrimination between physical limits, sensorimotor control limits, and cognitive limits. For instance, most people cannot flex the DIP joint independently of the PIP joint. This is a physical limit having to do with the kinematics of the tendons activating those joints. Drawing a perfect circle freehand is difficult. This is a sensorimotor limit that perhaps can be overcome with training, but still reflects the basic imprecision of untrained coordination, and not a physical or cognitive limit. Mastering a video game is difficult because of the physical coordination involved, but more so because of the many factors that must be considered to stay "alive"—the *cognitive load* is high. Cognitive limits come into play with tasks that involve complex coordinations or tasks that require more than one point of attention.

The lines blur, however, when previously difficult and cognitively complex tasks are mastered as skills requiring little or no conscious thought. This is exemplified in the hand as grips and abilities such as turning objects within the grasp, manipulating a pencil, or threading a nut on a bolt. It is hypothesized that skill acquisition is a process of integrating posture and movement patterns into lower-level sensorimotor programs that require only supervision from the higher-level brain functions (Keele, 1982). However, it is unknown to what extent the relations of the degrees of freedom affects this integration or what the parameters and limits are to this phenomenon. There may be some tasks too difficult to learn skillfully within a reasonable time frame.

With whole-hand input, it will be important to understand the parameters of degree-of-freedom coordination and skill development to create viable strategies of whole-hand control that maximize the degrees of freedom that can be controlled without exceeding constraints of proficient use.

4.2 Appropriate control design

Different control designs and implementation eccentricities will drastically affect the usefulness of whole-hand input. A well designed mouse, joystick, or tablet interface may easily outperform a poorly designed whole-hand interface. It will be important to know what aspects of tasks are suitable to whole hand control, what schemes and abstractions work best for implementing the hand-space to control-space mapping, and what aspects of the hand and its motion are important to monitor. Some of the more prominent issues are described below.

Rate control versus position control In many control designs a decision must be made to use rate control or position control. Each has its advantages, and disadvantages (Kim et al., 1987). Primarily, rate control allows manipulation of variables with infinite range or continuous cycles, such as 360° rotations, but must compromise between precision and speed of achieving a distant goal. This can be helped by using variable rate controllers, which still compromise precision with speed, only at a higher level. Position control allows both precision and speed based on the abilities of the human operator, but is best used locally. Large deviations from the center point must be accommodated by changes of scale, reducing precision.

Several excellent efforts have tried to overcome the compromises of rate and position control in the area of interactive searching of large data-bases using "fisheye views," and in computer graphics using logarithmic motion (Furnas, 1986; Mackinlay, Card, and Robertson, 1990). It may be possible to apply some of these techniques to whole-hand input control.

Distinguishing control motions from personal hand motions One of the issues particular to whole-hand input design is how the interface is able to distinguish between a command in the task domain and an unrelated gesture. The user must be able to disengage from the task, suspend input, or to rest. It is not desirable, for example, for a slaved robot to reorient a power-tool when the operator scratches his nose, or a surgical tool to continue cutting when the surgeon gestures to a nurse. The operator must be able to uncouple periodically from the task to perform non-task related motions or to rest. There are many ways this could be accomplished including foot switches, buttons, and other "dead-man switch" or clutch type controls.[2] A method that has been successfully tried with whole-hand input is a rapid hand motion such as a flick of the wrist (Sturman, Zeltzer, and Pieper, 1989). This motion is typically above the response frequency of the devices or motions being controlled, yet within the tolerances of the monitoring devices.[3] Other control schemes for controlling the response of whole-hand input are illustrated in the application prototypes in Section 10.

Ergonomics For whole-hand input, ergonomics refers to the comfort of a hand action and the risk of injury with extended use of a hand action. Repetitive motion injuries, such as carpal tunnel syndrome, can be severely debilitating and must be avoided in the design of whole-hand input methods. The medical and rehabilitation fields are replete with discussions of hand motion injuries and are a good source for this information, e.g., Hunter et al. (1984).

Skill The level of skill required for proficient use is another important issue. Some applications can afford long training times (although short training times are always desirable), such as space missions or career tasks, while in others, perhaps with high worker turnover rates, long training times are infeasible. Whole-hand input designs must take into account

[2]Doctors who use voice-activated microscopes for micro-surgery have found that a foot pedal or proximity detector is necessary so that they can indicate to the microscope when and when not to respond to voice commands (Dr. Joseph Rosen, personal communication).

[3]Flicking the wrist, if performed repeatedly, can lead to wrist irritation and eventual injury. There are many factors that must be balanced in developing these techniques.

the skill necessary for chosen hand motions and probable training times for a particular task. Unfortunately, although dexterous skill has been studied in the literature (Ervin, 1988; Robinette, Ervin, and Zehner, 1987), information is lacking on the dexterous skill of the average population with regard to specific hand motions, separate from functionality. Some of the experiments proposed in Section 11 address the need for these statistics of hand action.

4.3 Appropriate device

Current whole-hand input devices have been designed with knowledge of human anatomy, knowledge of the range of motion of the hand, and common sense as to what joints and motions may prove useful to monitor. Since there is little collective experience with these devices, current designs try to be general and cover many eventualities. As whole-hand input is more carefully studied, these common sense, general purpose designs may give way to designs based on more carefully collected data. It may be the case that for the majority of successful whole-hand input techniques only a few select finger joints need to be monitored, or that the spatial and temporal resolution of the devices should be less at some joints and greater at others. It may be that joint measurement is secondary to fingertip placement, or to overall hand shape. For example, Poizner et al. (1983) found that tracking fingertip motion alone is sufficient for human interpretation of American Sign Language.

There are also trade-offs among sensing technologies and devices. No single device provides everything desired for whole-hand input, i.e., inexpensive, unconstrained, unencumbered, and unambiguous readings of hand shape, position, and motion. Each device or method has advantages and disadvantages. Image-based systems have the advantage of not requiring users to wear anything on their bodies, but have to deal with occlusions and nuances of lighting. Mechanical devices do not suffer these problems, but can be encumbering. Users may reject a device they need to wear, preferring instead a device they don't have to pay attention to, or at most, can hold or put their hand on. Specific designs may be more effective than generic designs, e.g., the Digital Data Entry Glove versus the VPL DataGlove for signed alphabets (see Section 5.2). Until a method is found to overcome the current limitations of whole-hand monitoring, an analysis of each task's requirements in terms of these trade-offs is needed to determine the most appropriate device for capturing the motion of the hand for that task.

An additional consideration in the design of whole-hand input devices is the role of kinematic and dynamic constraints. Conventional input devices, such as mice, trackballs, joysticks, and so on, have helpful physical characteristics that limit range of motion, improve stability, and support unused degrees of freedom. For instance, a tabletop supports the unused degree of freedom of height for mouse input, and joysticks, dials, and sliders should have a slight amount of damping to facilitate smooth tracking and to avoid overshooting target positions. Likewise, appropriate constraints can benefit whole-hand input methods and devices. For example, a whole-hand input device used for manipulating a slowly moving robot hand could mechanically damp the operator's finger motions to prevent rapid movement and improve stability. A task requiring extensive use of only the fingers may benefit from an arm or palm rest. None of the current whole-hand input devices provide constraints to the hand; in fact, they avoid constraining the hand in accordance with design goals of being general purpose devices.

4.4 Importance of constraints on degrees of freedom

Although constraints often reduce the flexibility and degrees of freedom that can be applied to a task, effective control may require the assistance of external constraints and reduction of degrees of freedom. When free-hand motion is constrained by eliminating unwanted degrees of freedom, the user is no longer trying to avoid motion in these degrees of freedom, and can better concentrate on the degrees of freedom that do need to be controlled. For example, lateral forces are irrelevant with a lever that only goes up and down. The user can be sloppy in terms of pushing it from oblique angles. This allows fast, yet precise adjustments.

Constraints can also support unused degrees of freedom, helping to steady the degrees of freedom being used and reduce user fatigue. A tabletop provides this function to mouse input. The height of the mouse off the table is an unused degree of freedom. Resting the mouse on the table frees the user from always having to hold the mouse and steadies the hand's motion. For dials, a surface for the heel of the hand to rest on is important to provide a stable base for the finger motions. This also holds true for precision tablet work.

Although external restraints and supports of the hand are important for whole-hand input, the issues are complex enough to warrant separate study. The implementations of whole-hand input in this dissertation use free-hand motion—motion free from the effects of external forces and contact. Using the design method in Section 6, interface builders can see where tasks and input methodologies would benefit from external hand constraints.

4.5 Sensory feedback

Feedback will clearly play an important role in the use of whole-hand input techniques, however, the scope of the dissertation cannot do justice to the complexities of this phenomena. This section discusses the issues and research in the area of sensory feedback, particularly tactile and kinesthetic feedback. Beyond this, the dissertation focuses on the input side of human-computer interaction and does not study the effect of sensory feedback to the use of whole-hand input. Suggestions for future work (Section 11.5) include how the design method presented in Section 6 might be extended to include the effects of sensory feedback.

Visual feedback is of primary importance and only music systems seem to do well without it. The role of visual feedback is so widely studied and well covered in the literature that detailed discussions are better left to other sources. The issues relevant to whole-hand input include spatial cues (Goldstein, 1989; Kim, Tendick, and Stark, 1987), point of view (Ware and Osborne, 1990), and spatial and temporal resolution (Rogowitz, 1983; Rolfe and Staples, 1986, chapter 7).[4]

Tactile and kinesthetic feedback can have an important influence on manual task performance. Sensorimotor actions rely on appropriate feedback from cutaneous and musculoskeletal sensors, and reaction time from kinesthetic input is faster than from visual input (Evarts, 1974). Studies have shown that kinesthetic feedback can enhance task performance for many applications, but little is known as to what qualities of kinesthetic and tactile feedback affect performance (Brooks et al., 1990; Chin and Sheridan, 1989; Kilpatrick, 1976; Minsky et al., 1990; Noll, 1972; Ouh-young, 1990).

Clearly, any device that the hand manipulates can be emulated by monitoring the hand freely pantomiming those same manipulations. For instance, moving the empty hand across

[4] Rogowitz (1983) and Rolfe and Staples (1986) provide excellent reviews of problems and solutions for visual feedback in computer graphic applications and simulations.

a table top imitates a mouse. However, the motion constraints imposed by the physical devices and their inherent kinesthetics can be an integral part of the control they provide. Free-hand pantomime may never replace the performance available from a well designed joystick.

Some performance characteristics have to do with the motion constraints imposed by the device, and some with the kinesthetic feedback arising from the constraints. The degrees-of-freedom constraints can be compensated for in software by discarding extra degrees of freedom of the hand. The kinesthetic feedback from physical devices is much more difficult to emulate, especially to the whole hand, and its absence may have a profound influence on task performance.

In the future, it will be important to know not only what tasks must be performed with and what tasks can be performed without kinesthetic feedback, but what is the nature of the feedback required. For instance, in controlling a manipulator gripping an object, the nature of the feedback could range from a vibratory buzz on the fingertips indicating contact, to full tactile sensation including the sensing of textures, contours, and edges of the object being grabbed. Patrick et al. (1990) have shown that vibrotactile display (vibrating the fingertip pads) provides good tactile cues and that merely the suggestion of contact may be sufficient for many manipulation tasks. At the other end of the spectrum, Jacobsen et al. (1990) have developed a complex force-reflecting master-slave arm/hand that realistically reflects robot arm, hand, and finger forces and motions to a human controller. Both their system and a smaller, desktop force-reflective system by Iwata (1990) provide feedback to only a few of the degrees of freedom in the fingers and do not provide cutaneous feedback beyond that of the pressure used to apply the forces. Although the sensory cues do seem convincing, neither system has been subjected to experimental analysis, nor has the extent to which the force cues contribute to performance been studied.

Some of the mechanical difficulties with the development of good tactile and kinesthetic feedback to the hand include the difficulty of providing sufficient force to the finger joints without encumbering the hand with heavy actuators or networks of transmission lines or tubing. If "suggestions" of feedback can be used, the force requirements on the hand could be significantly reduced, simplifying the mechanisms (e.g., a buzzer on the finger to suggest contact may be simpler than a mechanism to provide an actual contact force). Another complication in providing feedback is that it is unclear how the different cutaneous and musculoskeletal receptors contribute to the sensations of touch and motion (Clark et al., 1985). Imprecise or incomplete feedback may be sufficient in some situations and grossly misleading in others.

As whole-hand input techniques develop, it will be important to know what tasks must be performed with, what tasks can be performed without, and what tasks can be performed with varying levels of kinesthetic and tactile feedback. Whole-hand input devices with which to perform these analyses may be on the market soon, as several companies are developing "tactile feedback gloves" (Stone, 1991; W Industries, 1991).

Auditory feedback is another viable form of sensory feedback for whole-hand input. Auditory feedback is expected in the use of whole-hand input for musical performance, but it is less standard in other applications. The literature comments on the usefulness of everything from clicks and bells, to changes in pitch and volume, to synthesized speech, designed to enhance task performance (Buxton, 1985; Gaver, 1986; Jones, 1989a). An advantage of auditory feedback is that it does not require shifting attention from visual or other channels of feedback.

As with kinesthetic and tactile feedback, auditory feedback can improve task performance, but the required nature of the feedback necessary for different tasks is not well understood. For example, in some situations a simple "click" may be an effective method to indicate the achievement of a target; in others, it may be insufficient or irrelevant. As with tactile and kinesthetic forms of feedback, it will be important to test the role of auditory feedback in the development of whole-hand input.

4.6 The use of gestural languages

The computer interpretation of gestures is a difficult task and is actively being studied by several researchers in the United States (see Section 5). The problems of interpreting language structures is beyond the scope of this thesis. The work here uses simple, unconnected gestures and signs. Nevertheless, interpreting signed language is an interesting and important problem that deserves further study.

4.7 "Point, reach, and grab"

Another aspect of whole-hand input covered only briefly in this dissertation, is what can be called the "point, reach, and grab" paradigm. This refers to a form of whole-hand input in which a graphic representation of the user's hand appears on the screen, duplicating the user's motions. The graphic hand can interact with other objects on the screen, allowing the user to manipulate those objects as if they had tangible existence in a physical world. Almost all whole-hand input devices currently being used are engaged in this context.[5] As a result, it can be considered "widely used," but not formally studied. "Point, reach, and grab" only touches the richness of whole-hand interaction as presented by this dissertation, and is discussed in more detail in Section 5 along with other work in the field.

4.8 One-handed vs. two-handed input

Most manual tasks are done with two hands, often one steadying the work of the other. One can foresee that whole-hand input will be no different, and that by using two hands instead of one, even more work can be accomplished. At least one research project (Buxton and Myers, 1986) has discussed the role of two hands in computer input. For reasons of complexity and whole-hand device availability, this dissertation concentrates on the problems of one-handed input, leaving the issues of using both hands to future research.

[5] VPL Research actually received a patent relating to the use of a computer graphic hand controlled by an instrumented human hand where the graphic hand is capable of interaction with other graphic "virtual" objects (Zimmerman and Lanier, 1991).

5 Background

The concept of whole-hand input is not new, but only in the last five years, with the introduction of affordable whole-hand input devices, have researchers begun in earnest to develop applications using free-hand motions as input. Much of the existing work examines the use of specific whole-hand input devices in the context of specific applications. Few researchers have examined whole-hand input in general, or comprehensively addressed the issues related to its development. Consequently, this dissertation has no dominant precedent in any one field and borrows from several domains of study. This section discusses prior and related work, providing a technical and historical backdrop to the dissertation.

Some of the application-specific related work has been discussed in Section 3 along with the general use of whole-hand input in those application fields. The current section first describes previous and current work addressing the general topic of human-computer interaction, and then work that has been done with specific whole-hand input devices.

5.1 Human-computer interaction

The field of human computer interaction (HCI) can be broken down into three main levels of study: the theoretical or psychological level, the device interface level, and the psychophysical level. Extensive research has been done on each of these levels. Unfortunately, the complexity of human behavior makes HCI a difficult area in which to validate theories. Results tend to be context dependent (although, less so at the psychophysical level).

In an interesting essay on this topic, Carroll and Campbell (1988) argue that the artifacts developed in HCI—the devices, techniques, and systems—embody theories, but the theories they embody are not powerful enough to guarantee success in other applications. They claim that many artifacts may not be reducible to explicit theory and may be incomprehensible apart from the situations in which they are used. Thus, they term HCI a *design science*. This makes reliance on prior work tenuous because, by its paradigmatic nature, each theoretical work is uniquely inseparable from the specific application generating it.

Good HCI models can be appropriated from previous work, but it is difficult to use theories about why the models are successful. This is not to say that prior work is irrelevant, but that where scientists habitually seek universality through theories, universal application is difficult to abstract from HCI theories. Thus, in the study of whole-hand input, one must be aware of the fragile nature of theoretical work. Caution must be taken in translating results to other contexts.

In a related report, Carroll (1988) describes how scientifically rigorous psychological approaches to HCI, involving testing of low-level phenomena, have had little practical impact when expanded to general application. On the other hand, he says, attempting only to formulate models of the user's mind and actions ignores the important human-factors aspect of HCI.[1] The conclusion to be reached is that environment, task, device, and human factors must be integrated for practical HCI development.

Theoretical level Several concepts important to whole-hand input have been established at the theoretical or psychological level of HCI. In the field of ecological psychology, researchers believe that the psychology of human computer interaction must be studied in terms of the human-task environment in which the actions occur. Flach (1990) provides a good introduction to the HCI implications of ecological psychology theories. Vicente and Rasmussen (1990) describe *ecological interface design* as a process of extracting features of tasks at various levels so as to make the geometry of the interface reflect the nature of the task in a way that exploits direct perception. One of the goals of ecological interface design is to allow operators to act directly with the task, making the intermediary sensor displays as functionally transparent as possible. Since transparent (or natural) interfaces are one of the goals of whole-hand input, ecological interface design may provide some guidelines for the analysis and mapping of tasks to whole-hand input.

Another important concept is that of *direct manipulation* (Shneiderman, 1982; Shneiderman, 1983). This is where the user experiences interaction as being directly with the objects of interest rather than through an intermediary system. The Apple Macintosh operating system uses direct manipulation for most of its operations. To move files from one directory or folder to another, a person clicks the mouse cursor on the files to be moved and drags them to the new folder. In a traditional operating system a command is typed to the operating system and (conceptually) *it* does the operation. Hutchins, Hollan, and Norman (1986) talk about *directness* as an impression or feeling about an interface resulting from the commitment of fewer cognitive resources. The more a person has to think about an interface, the more a person feels removed from the task. They describe

[1] Carroll references Whiteside and Wixon (1987), and Winograd and Flores (1986).

distance as "the gulf between the user's goals and the way they must be specified to the system." These concepts provide a useful context with which to view some of the goals of whole-hand input.

Expanding on the theme, Laurel (1986) describes the "computer-as-a-tool" mode of interface as an artifact of the evolution of interface design. When people use a computer, she says, they are interested in the application, not the use of a computer. Therefore, the computer should become transparent. The interface should take on the aspects of the task and the user should become an operator, or agent, in the domain of the task, rather than a distanced observer working through an intermediary operating system and command structure. Wixon and Good (1987) also support the notion of transparency and argue that "transparency" and "support for breakdown" should be used as measures for the usability of computer systems. They claim that schemes of hierarchal categories of user interface are misleading, and that most computer systems have a continuum of use and modality that crosses category boundaries. They conclude their essay with the hope that designers and researchers will think along continuous dimensions of usability rather than rigid categorizations.

Device interface level At the device interface level, the most important general developments have to do with systems of describing virtual input devices and taxonomies of input devices. Foley and Wallace (1974) described input tasks so as to be independent of device. Their purpose was to allow the discussion of input models without the dependency of hardware technologies. They classified four virtual devices, the *pick*, the *button*, the *locator*, and the *valuator*. This since has been refined and integrated into the GKS system (Enderle, Kansy, and Pfaff, 1984) as *pick*, *choice*, *locator*, *valuator*, *stroke*, and *string*. These categories can be used to provide a device-independent input library, but do not take into account the properties of specific devices that make them suited for a particular task. A trackball, tablet, or mouse can be used as a locator device, but provide different levels of performance depending on the task.

In an effort to address the human factors of devices, Buxton's *Taxonomy of Input Devices* (Buxton, 1983) categorizes input devices in terms of properties sensed (position, motion, or pressure) and degrees of freedom. Card, Mackinlay, and Robertson (1990) have improved upon this taxonomy by including both the continuous and discrete properties of input devices. Buxton (1990) has taken the next step and proposed a model which accounts for the hybrid discrete/continuous properties of devices. This model uses state changes to describe input sequences and relates tasks and devices using these state descriptions.

There have been many studies of specific devices and human performance. However, as has been stated earlier, the majority of them are too context dependent to be generally useful. One early experiment stands out. Most researchers agree on the validity of *Fitts Law* (Fitts, 1954), or variations thereof. Fitts tested the time it took people to accurately move small objects from one point to another.[2] He found that target acquisition times have a logarithmic relationship to the size and distance of the target. This is expressed formally as

$$MT = a + b \ \log_2 \frac{2A}{W}$$

where MT is movement time, A is movement amplitude (distance between start and finish), W is target width (size), and a and b are application and device dependent constants. Card, English, and Burr (1979) confirmed Fitts Law for two-dimensional computer interaction (selecting text on CRT displays), and found values for a and b for different devices and tasks. When similar tests are brought to three-dimensional computer input, results are less conclusive. Researchers have found that display styles (such as stereo versus perspective views) and input metaphors affect the results (Beaten et al., 1987; Ware, 1990; Ware and Osborne, 1990). This may have to do with the increased cognitive load of correlating the two-dimensional screen image (or synthetic stereoscopic image) with the subject's mental model of the three-dimensional space.

Psychophysical level Psychophysics is an active field that covers a wide range of studies. The areas relevant to whole-hand input have to do with the sensorimotor control of the hand and arm. Excellent reviews of the sensorimotor control field can be found in Pew (1974) and Kelso (1982). Of specific relevance to whole-hand input are discussions of the problems of managing degrees of freedom (Turvey, Fitch, and Tuller, 1982), coordinated control (Tuller, Turvey, and Fitch, 1982), and space-time invariance of certain motor skills, such as hand-writing (Viviani and Terzuolo, 1980). Other studies give suggestions as to the low-level importance of tactile versus visual feedback (Evarts, 1974). In the area of

[2] In his experiments, Fitts had subjects move a pen point between two rectangles, transfer washers between two pegs, and transfer pins between sets of holes. He measured the time it took to reach the target point, based on target size and movement distance.

clinical analysis of hand function, relevant studies include low-level analysis of simple finger motions (Cole, Gracco, and Abbs, 1984; Cole and Abbs, 1986), and range and ability of finger motion (An et al., 1979; Becker and Thakor, 1988; Chao et al., 1989; Mesplay and Childress, 1988).

5.2 Whole-hand input devices

Camera-based devices

For many years, biomechanics labs across the country used LED-based systems, such as Selspot by Selcom or OptoTrak by Northern Digital, to track the motion of the body and limbs (Mann and Antonsson, 1983). These systems use multiple infrared cameras focused on the subject wearing LEDs activated in sequence. A computer system analyzes the position of each LED in each camera's visual field and calculates the world-space position of the LED. These systems are limited by the computer time needed to calculate the world-space position, occlusions of the LEDs by the body, lengthy calibration procedures, and positional accuracy.[3] Nevertheless, LED systems have been used successfully as tools for clinical analysis of body movement.

In the early 1980's researchers at the MIT Architecture Machine Group and then the MIT Media Lab used a camera-based LED system to track body and limb position for real-time computer graphic animation (Ginsberg and Maxwell, 1983; Purcell, 1985). The LED position data was sent to a computer graphic rendering system which drew a representation of the user's body, mimicking the user's motions. This work included a glove studded with LEDs to track finger motion. Hall (1985) mentions using the LED glove in an experimental system that performed table-lookup on finger postures to allow input by finger spelling. This simple system begins to lead into the use of the hand for signed language, but other than this, no attempt was made by them to interpret finger or hand motions.

Poizner and other researchers at the Salk Institute in La Jolla, California, also used a camera-based LED system to analyze signed language. In 1983 they reported on their research to analyze hand motions of American Sign Language (ASL) using point light displays (Poizner et al., 1983). They placed the LEDs on the hand and arm so as to minimize

[3] Mann and Antonsson (1983) were able to get the positional accuracy of the Selspot system to 0.1 percent of the visual field; sufficient for limb movement, but not for fine finger motions.

occlusion during signing. Analysis was done in non-real-time after the motion data had been collected. This avoided some of the computational speed problems usually associated with moving point light displays. They proposed various analytical techniques, including feature analysis and frequency analysis, from which to qualify the linguistically relevant features of signed language. Although their interest was in understanding the phenomena of signed languages, their work can be adapted to computer understanding of a gestural lexicon or gestural control. Of special relevance are their methods of motion analysis to derive useful metrics of signing and their mapping of signs into various dimensions of a visual-articulatory space. They contend that humans can articulate and interpret hand motion along these dimensions. By using these same dimensions in gestural control, perhaps complex (i.e., powerful) yet manageable methods for gestural control can be developed.

There has been comparatively little other work in capturing hand motion using camera-based systems. The main problems with image-based visual tracking of the hands are that the resolution of conventional video cameras is too low to both resolve the fingers easily and cover the field of view encompassed by natural hand motions; the 30 (or 60) frame per second conventional video technology[4] is insufficient to capture rapid hand motion; fingers are difficult to track as they occlude each other and are occluded by the hand (a common occurrence); and computer vision techniques are not developed enough to sufficiently interpret visual fields in real-time. For these reasons, researchers have turned to mechanical systems for practical monitoring of hand motion. There is reason to believe that when the problems of camera-based systems are overcome, there will be a return to this method of capturing hand motions. It provides the user with the convenience of not wearing devices or special clothing, or otherwise being distracted by the monitoring equipment.

Two camera-based “clothing-free” systems have survived. One has been developed by Myron Krueger and the other by Vivid Effects in Toronto, Canada. Both systems use silhouette images of the user. Neither deals with the problems of occlusion and image-merging of fingers close together.

Myron Krueger’s systems are constructed to allow people interaction with computers without the need of encumbering equipment (Krueger, 1990). By using custom hardware to process the silhouette images, he overcomes some of the usual image processing speed

[4]Infrared systems, such as Selspot can operate above 300 Hz, and special-purpose high-speed video cameras are available; but conventional video cameras are limited to 60 Hz.

problems. His techniques are successful at recognizing parts of the body—head, legs, arms, fingers—if they can be seen in the silhouette. In one application, participants can draw figures with their fingers. When the computer sees that the thumb and index are outstretched on both hands, it draws a curve that inscribes the region between the two hands (Figure 6). The size and shape of the curve can be changed by moving the hands or fingers. A rapid pull away from the curve fixes it in place on the screen.

One of Krueger's goals is to develop an entire computer-based workspace that requires a minimum of mechanical devices, instead relying on vision techniques to interpret the user's hand and body motions. He sees the main limitations of his system as spatial and temporal video resolution, and separation of foreground from background in cluttered environments.[5]

figs/krueger.eps

Figure 6: *Manipulating graphics by hand* Fingertips control a Bezier curve. (From Krueger, 1983, p. 146.)

Vivid Effects has commercialized an Amiga-based system that also processes silhouette and video images and chroma-keys them into optical disk based video material and computer graphics (Vivid Effects, Inc., 1989). Their software triggers events such as playing of sound, mode changes, and graphics animation, when the user "touches" trigger points in the recorded image. Various versions of the system allow users to play virtual instruments, sing "with" a video rock-and-roll band, be a character in a computer graphic adventure game, and explore strange landscapes. The limitations of this system are that it does not "understand" parts of the body as Krueger's system does so that the parts of the silhouette cannot be differentiated.

[5] Myron Krueger, personal communication, 1990

Mechanical devices

Master-slave controllers Master-slave controllers that connect hand and manipulator motion through mechanical, hydraulic, and/or pneumatic linkages have been used for decades for handling hazardous materials (Minsky, 1980; Sheridan, 1989). These first manipulators afforded rudimentary dexterity, but served their function. As technology advanced, more sophisticated tasks were considered for teleoperation and dexterity requirements increased.

Recent developments of dexterous robot hands (Jacobsen et al., 1986; Mason and Salisbury, Jr., 1985) have attempted to raise the potential dexterity in telemanipulation to that of the human hand. These robotic dexterous hands are kinematically similar to human hands, and attempts have been made to control them with whole-hand masters (Hong and Tan, 1989; Pao and Speeter, 1989; Speeter, 1989). Others have concentrated on improving control through the use of kinesthetic feedback from the robot (Bejczy and Salisbury, Jr., 1983). Providing this feedback is a difficult problem (Chin and Sheridan, 1989; Durlach, 1989) and current implementations—such as those at the University of Utah—require large devices to accurately reflect the forces felt at the manipulator. A disadvantage is that these devices are too bulky to be used in many general applications.

In related work, Kilpatrick (1976) used the master side of a large force-reflecting master-slave manipulator arm to demonstrate the use of force-feedback in computer-aided task interaction. Later, Ouh-young (1990) used the arm to successfully assist biochemists in analyzing dockings for drug molecules. Both researchers found force to be a useful feedback tool.

Various companies and research laboratories are developing smaller force and tactile feedback devices. Some are mounted on a small base appropriate for desktop use and provide force-feedback to the position of the hand and fingers within a limited space (e.g., Iwata, 1990). Others incorporate force producing elements into gloves, providing tactile sensations while still allowing the arm free range of motion (e.g., Stone, 1991). Many of the small force-feedback devices are proprietary developments slated for commercial release and detailed information has not been published.

“Sayre” Glove DeFanti and Sandin (1977) reported on the development of an inexpensive, light-weight glove to monitor hand movements. Based on an idea from Rich Sayre of the University of Chicago, they used flexible tubes (not fiber optics) with a light source at one end and a photocell at the other. Tubes were mounted along each of the fingers of the glove. As each tube was bent, the amount of light hitting its photocell decreased evenly. Voltage from each photocell could then be correlated with finger flexion. They found this to be an effective method for multi-dimensional control.

Digital Data Entry Glove In 1983, Gary Grimes of Bell Telephone Laboratories received a patent for a glove interface for the entering of ASCII data (Grimes, 1983). The patent covers the use of a special electronic glove whose sole purpose is to interpret a manual alphabet for digital data entry—a keyboard replacement. The glove itself is made of cloth on which is sewn numerous touch, bend, and inertial sensors, specifically positioned so as to recognize the Single Hand Manual Alphabet for the American Deaf. The circuitry of the glove is designed so that unique combinations of sensor readings cause the output of 80 of the 96 printable ASCII characters (a superset of the Single Hand Manual Alphabet for the American Deaf).[6]

VPL DataGlove™ Zimmerman et al. (1987) developed a glove that monitored ten finger joints and the six degrees of freedom of the position and orientation of the hand. The DataGlove (as it was called) was an improvement over existing camera-based hand-monitoring techniques because it operated faster and did not rely on line-of-sight observation. It was better than previous master-slave manipulators because it was light-weight, comfortable to wear, unobtrusive to the user, and general purpose.

Commercialization of the DataGlove by VPL Research, Inc. at a reasonable cost to research institutions has lead to its widespread use around the world.

In its current stage of development, the DataGlove consists of a lightweight lycra glove fitted with specially treated optical fibers along backs of the fingers. Finger flexion bends the fibers, attenuating the light they transmit. The signal strength for each of the fibers is sent to a processor which determines joint angles based on precalibrations for each user. Most DataGloves have ten flex sensors, one for each of the lower two knuckles of the digits,

[6] Grimes's invention has not been commercially developed.

but some have been made with abduction sensors that measure the angle between adjacent fingers. Position and orientation of the palm is determined by a Polhemus[7] sensor attached to the back of the hand, registering distance and orientation to a companion transmitter fixed in place nearby. The finger-flex accuracy is rated at 1° joint rotation but formal testing and personal observations have shown the actual flex accuracy to be closer to 5° (Wise et al., 1990). The DataGlove can collect finger data at approximately 60 samples per second.

Most of the DataGlove research has used the hand as a "natural" extension of the user into the computer environment replacing more conventional input devices but adding little or no new functionality. This is not to say that the DataGlove has no advantages over conventional input devices. It can provide a much more natural interface than a mouse or joystick. However, when viewed in terms of functionality, few have used it as more than a glorified three-dimensional mouse.

The developers of the VPL DataGlove have been primarily interested in simulated environments or *virtual realities*, and have used the hand as the user's manipulative extension into those environments (Kelly, Heilbrun, and Stacks, 1989). Users wearing the DataGlove in the VPL system see a graphic hand which follows the motions of their hand in the simulated environment. By pantomiming reaches and grabs, the user causes the graphic hand to reach and grab objects in the simulated environment. The viewpoint can be moved by pointing in the desired direction and "flying" to the destination.

The actual implementations of the grab and flight behaviors are based on software that triggers events on recognized finger postures.[8] Thus VPL's entire hand interface can be reduced to a set of abstracted input devices. The hand location is a *locator*, grabbing is achieved through posture recognition—a *button*, and motion through the environment by pointing your finger in the direction of travel is a *locator* and *button* combination. Functionally, the DataGlove could be substituted with a *bat* in VPL's application. A *bat* is a six-degree-of-freedom locator with one or more buttons (Ware and Jessome, 1988). The buttons are functionally equivalent to the posture recognition of the DataGlove software.

[7] This three-space sensor, made by Polhemus, uses low-frequency pulsed magnetic fields to sense the six degrees of freedom (three-space position and orientation) of a small sensor relative to a source transmitter. See (Raab et al., 1979) for technical details.

[8] VPL uses look-up tables containing min/max values which bracket the range of finger sensor values defining a posture (see Section 9.2). Following VPL's example, most researcher's DataGlove systems use similar methods, some with RMS or other error reducing techniques

It is perhaps a less natural interface to the environment, but it is functionally equivalent.

The Aerospace Human Factors Research Division of the NASA Ames Research Center studied the VPL DataGlove in its initial stages of development and used it for interaction with their Virtual Environment Display System (Fisher et al., 1986; Fisher, 1989). Like VPL, they used the DataGlove as a tool for grasping and moving objects, indicating direction of motion, picking from menus, and invoking system commands (by postures).[9] They also have used the location of the hand as a trigger for various events such as drum beats in a virtual drum machine. Like VPL, their use of the DataGlove provides functionality equivalent to a bat, but profits from the naturalness of using the hand.

In much of the literature, the DataGlove is used similarly to its application at VPL and NASA. The hand's graphic image is displayed in an interactive computer environment and used as a tool for "point, reach, and grab" interaction. At the MIT Media Lab, work prior to this dissertation used the DataGlove as a master for a graphical hand in a virtual environment. The user could grab, move, and throw objects with the graphical hand, as well as use finger postures and motions to select from on-screen menus (Zeltzer, Pieper, and Sturman, 1989). Kaufman and Yagel (1989) used the DataGlove similarly in a modeling environment. The user could grab and manipulate objects on the computer screen. Feiner and Beshers (1990), and Takemura, Tomono, and Kobayashi (1988) also used the DataGlove to allow users to "touch," grab, and manipulate on-screen objects and recognized finger postures as event triggers (buttons).

The advantage of this model of interaction is naturalness—users' actions are closely correlated with those that might be performed on physical objects. However, in each of these applications, the DataGlove functioned little more than a bat.[10] In fact, in the MIT implementation, the function of the DataGlove could be substituted by a Spaceball™—a six-degree-of-freedom force input device with eight buttons. The interface to the Spaceball was similar to the interface to the DataGlove with button events substituting for posture recognition.

[9] VPL and NASA developed similar applications at the same time. Ideas were traded back and forth in an effort to develop the technology. To say that one copied the other would be misleading.

[10] The MIT group first considered implementing the virtual hand as a dynamic object in the simulated environment so that grabbing, pushing, and other interactions would be physically based. However, they did not have the computational power to implement this scheme in a real-time system. As an alternative, they approximated the functionality with posture recognition.

Although the Polhemus is a position-control device, while the Spaceball is a rate-control device (and thus affects the input task differently), the functionality of the two was the same, i.e., manipulating objects. The Spaceball does not allow the same level of coordinated three-space motion as the Polhemus (mounted on the DataGlove), but does perform better for tasks requiring precision location or steady motion. This is partly due to the difference between rate-control and position-control, and partly due to the inherent jitter of freehand motion and the susceptibility of the Polhemus to electromagnetic disturbances.

At AT&T Bell Laboratories, Weimer and Ganapathy (1989) used a DataGlove in the same way as the systems described above, except they implemented two thumb-based gesture controls called *clutch* and *throttle*. Clutching was used for incremental transforms, such as rotation. When the thumb was brought towards the index finger, the screen object followed the rotation of the hand. When the thumb was pulled back, the screen object did not rotate. With this clutch mechanism, object manipulations could be ratcheted, avoiding uncomfortable contortions of the hand and arm. Throttling was a variation of the clutch mechanism in which the abduction angle of the thumb was used to scale the effect of a hand motion. Their scheme can be described in terms of virtual devices by calling the clutch a button based on thumb posture, and the throttle a valuator based on the angle of the thumb.

Two research projects have used the DataGlove to control a Utah/MIT Dexterous Hand robot manipulator (UMDH). At AT&T, Pao and Speeter (1989) constructed algebraic transformation matrices to map human hand poses to robot hand poses. The transformation matrix was necessary to overcome the kinematic differences between the hand (as transduced by the DataGlove) and the UMDH. The user manipulated the UMDH by mimicking the desired poses. At NYU's Courant Institute, Hong and Tan (1989) resolved the kinematic differences between the human master hand and the robotic slave hand by determining the position of the fingertips of the user's hand and then driving the robot hand fingertip positions to match.

Takahashi and Kishino (1990) of the ATR Research Labs in Kyoto, Japan developed a coding scheme to allow computer recognition of the Japanese kana manual alphabet. They used the DataGlove to capture hand posture[11] and recognized signs through a combination of principal component analysis to determine the contributions of each finger joint to the

[11]Takahashi and Kishino used the term *gesture*. However, they did not look at hand motion, so the term *posture* is used here.

differences between signs, and cluster analysis to group hand configurations. Because of the difficulty of accurately measuring the lower thumb joint with the DataGlove and because some of the signs have similar finger positions they were able to discriminate only 30 of the 46 kana signs.[12]

Slightly more complicated is work by Fels (1990) using a DataGlove to interpret hand motion to drive a speech synthesizer. His particular approach used a three-stage back-propagation neural network trained to recognize gestural "words." He divided hand motions between finger positions and hand motion. Finger positions defined the root word while hand motions modified the meaning and provided expression. No finger motions were monitored, and hand motions consisted only of variable speeds of back and forth motion in the six three-space cardinal directions. His "language" was based loosely on conventional gestural languages and his study had more to do with using neural nets to interpret a lexicon of hand signs than with the process of communicating with gestures. Nevertheless, Fels reported a 92% success rate on the recognition of 203 signs based on 66 hand shapes combined with 6 gestures.

A drawback of using neural nets is that they require extensive training that must repeated from the start each time a new hand motion is introduced. Thus, this technique would be best used with fairly established lexicons.

In his report, Fels included an interesting analysis of hand-to-language mapping at various levels of granularity, from using hand motions for the control of parameters of an artificial vocal tract, to interpreting whole hand motions as words and concepts. The trade-offs, as Fels put it, are between extent of vocabulary—unlimited at the most granular level—versus ease of learning and speed of communication—highest at the word and concept level.

Although Fels demonstrated the viability of connectionist techniques for interpreting finger position and hand motion, it is uncertain if his techniques will hold up under the added complexity of finger motions. This will be necessary to interpret the full expression of signed languages. However, as a control structure for computer input, Fels's methods may be adequate.

Brooks (1989) also used a neural net to interpret DataGlove motion; in this case for robot control. Unlike Fels, Brooks incorporated dynamic gestures into the control language. He

[12] Recent information indicates that there is other Japanese work along these same lines.

used Kohonen nets[13] to recognize paths traced by finger motion in the n-dimensional space of the degrees of freedom of the digits. Since he had no Polhemus or other three-space tracking method, Brooks ignored three-space hand motion. Each Kohonen net (typically small—on the order of 20 cells) was trained to recognize a single gesture. By operating several concurrently on the DataGlove input, several gestures could be recognized. He achieved moderate success at simple gesture recognition, such as closing all the fingers, leading with the index; opening the thumb and first two fingers simultaneously; and moving from a neutral hand posture to a "pen" grasp posture. However, in his conclusion, Brooks stated that he has yet to show that his methods are sufficient for practical dynamic gesture recognition or that the DataGlove is an appropriate interface for robot control.

The three methods of hand shape and motion recognition described above (and the method used by Kramer, below) are conceptually similar. Basically, they analyze the hand-space-degrees-of-freedom vector for each posture or gesture, and match it to a landmark hand-space vector representing the target posture or gesture. The match must occur within error tolerances (usually Euclidean distance) weighted by the significance of each degree of freedom. In the Takahashi-Kishino method, the principal component analysis determines the weighting of the degrees of freedom. In Fels's neural nets this process is hidden in the coefficients for each node. Brooks's Kohonen net has few nodes, each with an n-space vector of coefficients. These coefficients contain the weightings, with the interaction between the nodes of the net determining the identity of a dynamic gesture. (Kramer's implementation, described below, uses a method similar to the one used by Takahashi and Kishino.)

Exos Dexterous HandMaster™ In 1987 Arthur D. Little, Inc. (ADL) in conjunction with Sarcos, Inc. developed a master controller for the Utah/MIT Dexterous Hand, a four-digit robot-hand (Marcus and Curchill, 1988). The controller was an exoskeleton-like device worn on the fingers and hand. Using Hall-effect sensors as potentiometers at the joints, it accurately measured the flexion of the three joints of each finger as well as the adduction of each finger and the complex motion of the thumb. Since the Utah/MIT Dexterous Hand has only four digits, the exoskeleton had no pinkie. After shipping several of the Dexterous HandMasters, Dr. Beth Marcus, the leader of the project at ADL, licensed the technology and formed her own company, Exos, Inc. After redesigning some of the mechanics and all of the electronics, Exos brought to market a five digit exoskeleton—the

[13] Brooks references (Kohonen, 1984).

Dexterous HandMaster, Series 2 (DHM).

The current version of the DHM measures 20 degrees of freedom—four for each finger, and four for the thumb. Based on initial experience, the accuracy of the device is well within 1° of flexion. A formal study found similar results with a 92 to 98 percent correlation between finger position and DHM readout, depending on the joint (Makower, Parnianpour, and Nordin, 1990). The DHM does not measure palm position or orientation, but a three-space sensor can be attached for that purpose.

The DHM is being used for clinical analysis of hand impairment as well as for experimental purposes in several research institutions. Speeter (1989) extended his work with the Utah/MIT Dexterous Hand and DataGlove to the DHM. Since the DHM is kinematically similar to the UMDH, the transformation matrix scheme used for the DataGlove is not necessary. Instead, Speeter transforms the raw sensor data into strings of 7-bit characters. Lexical recognition routines match string patterns to autonomous manipulation functions for the UMDH (similar to the poses used for the DataGlove).

The DHM has also been used by Tod Machover at the MIT Media Lab for controlling acoustic parameters in live musical performance (Machover, 1990). Section 10.4 describes the project in detail.

Power Glove™ Inspired by the success of the VPL DataGlove, the Mattel toy company manufactured in 1989 a low-cost glove for use as a controller for Nintendo games. The Power Glove, as it is called, uses flexible molded plastic on the back of the hand and fingers and lycra on the palmer side. Embedded in the plastic on the fingers are resistive-ink bend sensors that register overall flex of the thumb, index, middle, and ring fingers with two bits of precision each. Mounted on the back of the hand are sonar range finders (similar to those used in automatically focusing cameras) to locate the Glove in space accurately to 1/4-inch. The range finders also provide four bits of roll orientation for the hand (rotation of the wrist).

Although the least accurate of the whole-hand input devices, the Power Glove is also the cheapest by a factor of 100. It works with several pre-Glove Nintendo games, such as *Mike Tyson's Punch-Out* where punching motions control the swing of an on-screen boxer. Some games have been especially designed for the Power Glove. *Glove Ball* is one that allows the player to "hit" or "grab and throw" a ball against tiles in a handball-like court imaged on the screen. These games are fun to play and make good use of the whole-hand interface. In addition, many researchers are experimenting with the Power Glove as a low cost alternative to the DataGlove for initial research into whole-hand input. Although a general purpose computer interface is not publicly available for the Power Glove, people have reverse engineered the electronics necessary for connecting the Power Glove to a computer's serial port (Eglowstein, 1990).[14]

Virtex CyberGlove™ James Kramer has developed a glove-based system at Stanford University to translate ASL into spoken English (Kramer and Leifer, 1989). A custom-made cloth glove has sewn into the fabric strain gauges to sense 16 degrees of freedom of finger and wrist flexion. Pattern recognition software maps the finger position into a "hand-state vector." When the instantaneous hand-state lies close enough to a recognizable state, the corresponding ASL letter or symbol is put on an output buffer. When a phrase is complete, a special sign causes the result to be spoken by a voice synthesizer. Hearing-able participants in conversations type answers back on a hand-held keyboard. The first implementation of the system only interprets finger spelling, where each hand sign is a letter in the English alphabet. Further work is expected to recognize other sign-language gestures. Kramer plans to market the glove as the *CyberGlove* along with a CAD virtual environment through a start-up company, Virtex.

Space Glove™ W Industries is a British company marketing a virtual reality arcade game. In 1991 they released a glove dubbed the Space Glove™ for use with their Virtuality™ system. The glove is made of soft molded plastic that fits over the back of the hand. The fingers are placed through rings that sit between the PIP and MCP joints. The four MCP joint flexes are measured, as well as two flex angles of the thumb, all using sensors with 12-bit analog-to-digital converters. A three-space magnetic tracker is incorporated into the back of the glove. Personal experience in using the glove for a short period of time found it fairly responsive to MCP flexion and hand movement, but somewhat uncomfortable as the plastic rings around the fingers had little stretch and constricted the fingers. The stiffness of the rings also made it hard to get over the PIP joint when putting on or taking off the glove.

[14] Eglowstein's article has good descriptions and comparisons of the DataGlove, the DHM, and the Power Glove.

6 A Design Method for Whole-hand Input

Section 4 discusses the important issues in developing practical whole-hand input techniques. The primary questions revolve around the appropriate use, control schemes, and devices for whole-hand input. The *design method for whole-hand input* described in this section outlines a disciplined approach to addressing these questions for any chosen task.

The method is an iterative process in which the designer determines the feasibility of using whole-hand input for a particular application or set of tasks, and then analyzes possible whole-hand input techniques for each element of the application or tasks.

6.1 Synopsis of design method

The method is broken into several stages as shown in Figure 7.

Appropriateness In the first stage the application designer determines the appropriateness of an application for whole-hand input by asking a series of questions about the application. The questions are based on the salient features of whole-hand input: naturalness, adaptability, and dexterity. For example, consider the remote control of a space repair robot. The process of determining appropriateness would reveal that there are potentially natural ways to use the hand in the application, taking advantage of kinematic correspondences between the robot's appendages and the hand; that there are many different tasks and sub-tasks to switch between; and that dexterity would be useful in some of the complex manipulations that need to be performed. (In other cases the questions may reveal that whole-hand input is not the most appropriate method of interaction and that conventional devices should be used.)

Taxonomy The taxonomy categorizes styles of interaction for whole-hand input. Based on the application, specific styles of input can be chosen. These help guide the designer to specific models of using the hand. In the repair robot example, two promising styles of input would be continuous, direct control of the robot from motions of the hand, and the use of hand signs to indicate high-level commands and mode switches.

figs/design-method.eps

Figure 7: *Design method for whole-hand input* The design flow for developing whole-hand input for any specific application or set of tasks.

Evaluation guide Next, the application's tasks are decomposed into task primitives. For example, the task of replacing an electronic module on a satellite may require the primitives of unbolting an access hatch, removing the hatch, removing the defective module, storing it away, unpacking the new module, inserting it into place, and replacing and rebolting the access hatch. Each of the task primitives are analyzed with specific measures that quantify it in ways that can be related to analogous measures of hand actions. For example, unbolting the access hatch requires three degrees of freedom (two coordinated for holding the tool square to the bolt, and a third for unscrewing the bolt), six bits of precision in each degree of freedom of the tool orientation, and a modicum of steadiness to keep the tool square to the bolt.

The designer chooses likely candidates for whole-hand action based on the whole-hand input or application literature, previous experience, or direct observations of the ways the hand is currently used in the task or in similar tasks. Applying the evaluation guide's measures to the hand actions, the designer directly compares, measure for measure, the task primitive and hand action. For example, the lower joint (MCP) of the index finger has two degrees of freedom that could be used to position the tool on the bolt. The precision and steadiness of the finger are ample for the task primitive.

If hand actions do not meet the task primitive requirements, then they can be refined or new ones chosen based on correcting specific deficiencies revealed by the analysis.

Device selection When the designer is satisfied with a matching of hand actions to task primitives, a set of device capabilities (with similar measures to the evaluation guide) is used to select a whole-hand input device appropriate to the tasks and the chosen whole-hand input methods.

Testing and evaluation Finally, the input methods are tested in an application simulation or in the application itself. If necessary, the designer goes back to the evaluation guide (or even the taxonomy) to further refine the task description and whole-hand input methods, repeating this process until a control scheme is devised that meets the task requirements.

Completed interface The completed interface occurs when the designer is satisfied with the performance achieved with the set of hand actions chosen for the tasks. The completed interface should not be static and should be adjustable to individual user preferences.

The following sub-sections describe each of the steps of the design process in detail. Following this, Section 7 shows how three example tasks are treated with the design method.

6.2 Appropriateness of whole-hand input

The first step of the process is to determine the appropriateness of the application or set of tasks to whole-hand input. Appropriateness is assessed by answering a series of questions based on the features of the hand that make it a useful input device. These are the features described in Section 2: naturalness, adaptability, and dexterity. The main question the designer is trying to answer here is,

> *"Is the use of whole-hand input appropriate and beneficial to the application tasks?"*

The specific questions fall into four categories: naturalness, adaptability, coordination, and real-time control. The more positive the answers to the questions are, then the more whole-hand input is recommended for the application.

Naturalness

There are four ways tasks could be considered to benefit from the naturalness of whole-hand input. The question to be asked is,

> *"Are the following characteristics useful for controlling the tasks?"*

1. *pre-acquired sensorimotor skills*: these include the use of fine motor control, skilled use of the dexterity of the hand, and use of skills of everyday living (such as grips, and manipulation of objects). Taking advantage of pre-acquired sensorimotor skills can reduce the learning time and the cognitive load of a task. Examples: controlling

an acoustic mixing board with finger flexion, master/slave of a robot arm, drawing with the finger, puppetry.

2. *existing hand signs*: if there is an established lexicon of hand signs (postures and gestures) used in the tasks, these can be used with whole-hand input, reducing the learning time of existing operators. Commonly, these hand signs are codified versions of natural hand signals, often miming the desired action, and so lend to the naturalness of the task control. Examples: aircraft taxi control, crane operation, using the signed alphabet for communication or command selection.

3. *absence of intermediary device*: one feature of whole-hand input is the fact that although the user can be wearing a device, the user does not think in terms of manipulating the device to achieve the desired control, but experiences acting directly in the task. This increases the sense of presence, and can improve performance and reduce training time for many tasks. Example: robotic control, puppetry, signed communication.

4. *task control maps well to hand actions (position and motion of hand)*: this mapping can be either kinematic or cognitive. Kinematic mappings are those in which the degrees of freedom of the tasks map well to the degrees of freedom of the hand. For example, master/slave control of a robot hand, or rotation of the hand indicating attitude of a vehicle. Cognitive mappings are those in which hand motions easily can be imagined as physically controlling the tasks. For example, the flex of the fingers indicating pressure of a robot grip, or opening and closing of the fist controlling the action of a pump. The existence of good mappings reduces the learning curve and can increase operator efficiency.

Adaptability

Adaptability alludes to the number of different control models and structures the user has to move between to accomplish the tasks. There are two questions that need to be asked here:

1. *Are diverse modes of control used in the tasks?*
   If there are diverse modes of control, then the use of whole-hand input for a variety of the functions may allow operators to switch between control modes quickly and

smoothly, increasing the overall efficiency of task control. Another potential benefit of whole-hand here is that only one device, the whole-hand input device, is necessary for the different control modes of the application.

2. *Is it important to be able to switch between modes of control rapidly, and smoothly (i.e., with a minimum of distraction from the tasks)?*
   This question follows from the one above. If there are diverse modes of control, then is it important to be able to move between them efficiently? If the answer to this question is negative then the application will not take advantage of the adaptability of the hand in switching modes (except as it applies to reducing the number of physical input devices).

Coordination

Coordination of many degrees of freedom of a task increases the cognitive workload. Manual dexterity allows operators to coordinate degrees of freedom to reduce the cognitive workload. Thus, if a task requires the coordination of many degrees of freedom, whole-hand input techniques can be used to allow the hand's natural dexterity to reduce the complexity of the control. The question to be asked is:

1. *"Do the tasks require the coordination of many degrees of freedom?"*

Real-time control

By itself, real-time control is not an argument for whole-hand input; dials, or a mouse can provide real-time control in the absence of the other criteria. However, in conjunction with the other criteria, the requirement of real-time control can increase the value of using whole-hand input with an application. This is due to well-developed human eye-hand and hand-hand feedback loops allowing for both smooth steady correction to continual fluctuations in the system or environment and rapid and accurate response to sudden events. There are two aspects to real-time control. The question to be asked is,

"How important to the tasks are the following?"

1. *continuous monitoring*: this is where a task needs continual monitoring and adjustment, such as steering a car. This is in contrast to supervisory control where a task is performed by an automatic mechanism with occasional input from the human operator.

2. *rapid user response*: this is where a task may or may not need continual control, but the user must be able to intervene at any time and immediately affect a task, such as stopping a motor, or closing a valve.

6.3 Taxonomy of whole-hand input

The taxonomy of whole-hand input describes styles of use of whole-hand input. It is derived from the interaction between a categorization of *hand actions* and the *interpretation of hand actions* by a task. Hand actions are defined as position, motion, and forces generated by the hand. The interpretation of hand actions is the functional interpretation made by the user and/or the application of the hand actions. The categories of hand actions are orthogonal to the interpretation of hand actions. The interaction of these two aspects of whole-hand input fully describe the possible styles of use of whole-hand input. A single application may use many styles of interaction. The taxonomy helps to discriminate these styles. Styles may be closely identified with or best suited for specific whole-hand input techniques, so that determining a preferred style of interaction helps to guide the process of choosing the best whole-hand input techniques for a task.

The style of use of whole-hand input relates closely to the level of control the user has over an application, ranging from low-level tight-looped continuous (direct) control to a high-level supervisory control (Sheridan, 1987). At the lowest level, the control is continuous. Every action of the operator is communicated to the task and the operator must constantly monitor the task. For example, an operator using finger flex to control robot arm flex is using direct control. At the highest level, the operator is acting as a supervisor, intermittently instructing an agent performing the task. An operator acting at the supervisory level might point to the object to be grabbed, form the correct posture, and the robot arm would reach for and grab the appropriate object. At an intermediate level the operator might specify a direction for the robot arm to travel, but not the goal. The operator might at any time change the direction, but also can take no action and the robot arm will continue moving in accordance with the last instruction. At an intermediate level, the operator must attend the task more closely than at the supervisory level, but does not need to continuously control the task as is the case at the continuous level.

Hand actions

Position and motion of the hand can be organized into two broad classes of description: *continuous features*, and *discrete features*.

A. Continuous features – These are continuous quantities derived from the degrees of freedom of the hand. There are three subcategories within this description.

1. **Degrees of freedom** – This is the most basic category. Position and motion are described in terms of the raw degrees of freedom of the hand. The hand (excluding the wrist) has 29 degrees of freedom, 23 from joints on the hand (Figure 1), three for the free translation of the hand, and three for the free rotation of the hand (of course, these last six are a result of the degrees of freedom of the wrist, elbow, shoulder, and body). Not all of these are independent. For instance, many people cannot bend the DIP joint without also bending (or restraining) the PIP joint, and those that can bend the DIP independently only can do so while simultaneously hyperextending the PIP joint. This is a function of the DIP flexor tendons pulling across the PIP joints as well as the DIP joints (Kaplan, 1965; Tubiana, 1981).
2. **Derived features** – These are continuous quantities abstracted from the degrees of freedom of the hand. Examples include fingertip position, joint velocities, direction of motion, and volume enclosed by the fingers. Higher level continuous features and second order features can be abstracted from more basic features. For example, fingertip path, or relative velocity between fingertips.
3. **Forces generated by the hand** can be characterized by the normal and tangential forces on the 19 or 20 fleshy pads of the fingers and palm. Although this dissertation does not go into detail on the forces involved with whole-hand input, forces are included for completeness.

B. Discrete features – These are discrete characterizations of the hand actions and can be interpreted as input tokens. *Postures* are specific values or ranges of values of the degrees of freedom or subsets of the degrees of freedom of the hand, or specific values of the features of the hand. Examples of postures are full flexion of the finger joints and thumb (a fist), or a curled index finger touching the thumb-tip (American version of “okay”). *Gestures* are derived from motion of the fingers and hand, or from continuous features. Examples are the traditional wave of “good-bye,” “circling” (where the index fingertip traces a circular path), or the signs of American Sign Language (ASL).

Alternately, characterizations of discrete features may be based on functional descriptions. Grips, for example, are characterized functionally (hook, lateral pinch, power, precision, and so on) and are based on the characteristics of opposition and hand shape (Cutkosky and Wright, 1986; Mishra and Silver, 1989).

Interpretation of hand actions

Interpretation of hand actions is independent of the physical sensing device. It can range from a literal interpretation of the sensor values as degrees of freedom of the hand, to a complex synthesis of the sensor values into a signed language. Interpretation of hand sensor data has been divided into three broad categories. These were introduced in (Sturman, Zeltzer, and Pieper, 1989). The terminology used to describe the categories differs here from that first presented. However, the substance of the categories is similar. These categories can also be interpreted as conceptual control models. That is, they are a way of organizing the models people use for whole-hand input. Similar controls can appear differently depending on the conceptual model used.

1. Direct Interpretation – the user “reaches” into and interacts with the application with the hand as if the application consisted of physical objects being manipulated in the real world (Laurel, 1986). This also can be interpreted as a form of direct manipulation (Shneiderman, 1982). Most research using the DataGlove uses this model in conjunction with the mapped interpretation. Direct interpretation also refers to input modes where the hand mimics the actions of the object being controlled.

figs/direct.eps

Conceptually, direct interpretation of the hand is the simplest of the three basic interpretations. The user’s hand action is transferred directly, or through a mechanical or mathematical transfer function, to the action of an anthropomorphic manipulator or computer-graphic hand. The action of the hand is the action of the manipulator. Sometimes the action of the whole hand, including the five fingers, is used; sometimes only a subset is used, as is the case with the Dexterous HandMaster and the UTAH/MIT Dexterous Hand (Speeter, 1989). Direct interpretation is conceptually simple because the user can think in "natural" terms when performing manipulations. Computationally it is simple because the degrees of freedom of the hand are mapped directly to the degrees of freedom of the manipulator. However, in some cases, the mapping can be more complex; especially if the transfer function requires corrections for manipulator deviations from human hand kinematics, or if the behavior of the manipulator is nonlinear when compared to the human hand, as with the DataGlove and the UTAH/MIT Dexterous Hand (Hong and Tan, 1989; Pao and Speeter, 1989).

2. Mapped Interpretation – data from the hand device is mapped to the functions of a conventional virtual input device such as the GKS standard's pick, choice, stroke, locator, or valuator (Enderle, Kansy, and Pfaff, 1984); or mapped to continuous actions in the task domain that do not have a kinematic correspondence with the controlling hand joints.

figs/mapped.eps

Subsets of the degrees of freedom of the hand and features of the hand actions can be mapped to degrees of freedom and functions of more conventional forms of input. For example, the flexion of a finger joint can be used like a slider—a *valuator*, or hand shape (posture) can be used to signal events—a *button*. There are many different mappings which can be made. Any mapping or abstraction of hand action to emulate another device (usually of lower degrees of freedom) falls into this category. The taxonomy developed by Sturman, Zeltzer, and Pieper (1989) maps the hand shape and motions to the virtual devices *button*, *locator*, *valuator*, and *pick*.

3. Symbolic Interpretation – gestures and postures are interpreted as streams of tokens for a language which may vary from the stylized and limited lexicon of a traffic cop to the highly developed syntax of ASL.

figs/symbolic.eps

Symbolic interpretation can occur at several levels of complexity. At the most basic is the interpretation of hand postures as tokens in a syntax-free lexicon. This is the simplest to compute and can be accomplished by a variety of methods including look-up tables

(VPL Research, Inc., 1987), principal component analysis (Takahashi and Kishino, 1990), and neural nets (Fels, 1990). The next level includes the motion of the hand (gesture) in the interpretation. This complicates the recognition process since temporal data must be analyzed. Methods such as feature analysis (Kim, 1988), Fourier analysis (Poizner et al., 1983), and neural nets (Brooks, 1989; Fels, 1990) can be used here.

The most complex level includes the location of the motion with respect to the body (near the face, near the waist, to the side); the interaction between two hands; and derivative aspects such as speed, jerk, trajectory, repetition, and more. ASL has some fifty different grammatical processes that differ along eleven spatial and temporal linguistic domains (Klima and Bellugi, 1979; Poizner et al., 1983). All these must be interpreted to fully understand ASL. Once these have been recognized, they must be parsed into linguistic phrases. The syntactic parsing of language is a complex computer problem on its own, beyond the scope of this dissertation. It has been extensively studied in the artificial intelligence community. See Barr and Feigenbaum (1982) for references.

Even more difficult is free-form interpretation of hand motion in which there is little or no formal coding of meanings in hand motions. This requires an "intelligent" system that can analyze the context in which the hand motions are occurring and use common knowledge to infer the meaning of the user. For instance, the gestures for "come here" or "stop" can take on many forms, depending on the cultural, situational, individual, or immediate context. Humans are able to interpret hand signs in this way, and this is the manner in which most of us use hand signals. It may be some time, however, before computers are able to perform the same free association that humans depend on for free-form hand motion interpretation.

Actions/Interpretation

There are six broad combinations of hand action and interpretation that characterize styles of whole-hand input. They are:

	Direct	**Mapped**	**Symbolic**
Continuous	Continuous/Direct	Continuous/Mapped	Continuous/Symbolic
Discrete	Discrete/Direct	Discrete/Mapped	Discrete/Symbolic

Continuous/Direct: Continuous aspects of hand action generate signals which control kinematically similar actions in the task domain. This usually requires dexterous manipulations on the part of the operator. Examples: finger joints control robot joints, graphic hand follows motion of user's hand, attitude of aircraft matches rotation of hand.

Continuous/Mapped: Continuous aspects of hand action are mapped to logical input devices whose signals are mapped to arbitrary functions in the task domain. There is often no kinematic correspondence between the hand action and the task primitive. This style also requires dexterous manipulations on the part of the operator. Examples: Finger joint rotations control the facial expressions of an animated character, speed of hand motion changes speed of device rotation, fingertip position locates cursor.

Continuous/Symbolic: Continuous aspects of hand action are interpreted by the system to determine the operator's intention. This may require knowledge-based reasoning on the part of the system since there is not a one-to-one correspondence between hand action and system response. The difference between continuous/symbolic and discrete/symbolic styles is that in the former, the system managing the task determines the instruction based on continuous signals from the whole-hand input system, while in the latter, the whole-hand input system decodes the instruction and transmits it as single token to the task system. Example: waving the hand or fingers in a particular direction to indicate motion in that direction. The system determines how best to achieve that motion, and/or which motion is being referred to.

Discrete/Direct: Discrete aspects of hand action (generally postures) correspond to similar configurations in the task. There is probably a narrow use for this style of control since there seem to be few applications that are able to use it. The salient example was developed by Hong and Speeter for mapping DataGlove and DHM positions to robot hand position. Discrete DataGlove/DHM postures were mapped to similar discrete Utah/MIT hand configurations. The user, in effect, signaled the hand as to which pre-determined configuration to form by mimicking the posture with their own hand. The actions taken by the system in these discrete categories are predominantly rule-based. The system recognizes a discrete hand action and performs a pre-defined function based on a rule set associating the function with the hand action. Alternately, discrete/direct can be seen as a snapshot of the hand state;

released either periodically (e.g., every five seconds), or functionally (e.g., when the hand is still for two seconds).

Discrete/Mapped: Discrete aspects of hand action indicate discrete levels of activation in the task domain. Examples: Gear turns as long as fingers are closed, current is maintained as long as hand is in motion, number of fingers closed indicate four levels of force to be applied, circling the index finger indicates hoisting the load on a crane.

Discrete/Symbolic: Discrete aspects of hand action generate commands to the application. These commands typically select, invoke, or terminate system functions. Symbolic interpretation of hand actions involves either rule-based procedures or knowledge-based reasoning on the part of the system. In the case of the former, pre-programmed actions occur based on the recognition of specific postures or gestures. There is a one-to-one mapping from hand action to task response. Examples are: closed fist indicates all action to halt, lights go on when index finger is flicked upwards, flicking the wrist changes modes. In the case of knowledge-based processing of the gesture or posture there is no one-to-one mapping of hand action to task response. The task system must determine the intent of the hand action. Most of our everyday hand signing falls into this category.

6.4 The evaluation guide

There are infinite variations of hand motions that can be used as input to an application. To guide the designer in choosing the best whole-hand control for a task, a set of measures have been specified to be analogous in the task and hand-action domains. By comparing the measures as applied to the task to the measures as applied to the hand, the designer can iteratively refine whole-hand input strategies on a measure-by-measure basis.

The measures in the evaluation guide cover a broad range of categories. Some are quantitative and can be expressed as single valued variables or sets of variables. Others are more subjective and reflect a combination of qualities. *Task characteristics and requirements*[1] describe the task primitive to be performed. *Hand action capabilities* characterize the control available from hand actions.

[1] For brevity, here and in subsequent discussion of the evaluation guide, the term "task" also refers to "task primitive." When significant, the distinction will be made.

The measures are as follows:

Task characteristics and requirements	**Hand Action** capabilities
1. Degrees of freedom	1. Degrees of freedom
2. Task constraints	2. Hand constraints
degrees of freedom	range of motion
physical constraints	coupling
temporal constraints	spatial interference
external forces	strength
3. Coordination	3. Coordination
4. Resolution	4. Resolution
spatial	spatial
temporal	temporal
5. Speed	5. Speed
6. Repeatability	6. Repeatability
7. Steadiness	7. Steadiness
8. Endurance	8. Endurance
9. Expressiveness	9. Expressiveness
10. Modality	10. Adaptability
11. Task analogy	11. Familiarity
comparison to existing methods	similarity to existing skills
similarity to other tasks	similarity to everyday motions

Procedure The designer starts by decomposing the application task into task primitives. For instance the task of stacking blocks has primitives that include grasping, moving, orienting, and releasing. Sometimes task primitives are linked and should be analyzed together. For example, if moving a block is conceptually or mechanically coupled with orientation as with a robot arm, then the two should be treated as one task primitive. Conversely, if moving involved one mechanical device, such as a conveyor belt, and orientation involved another, such as a turntable, then the two should be analyzed separately.

As each of the task primitives are quantified using the evaluation guide, the designer selects potential hand-actions for the task primitive. This choice may be based on literature reviews, prior experience, or direct observation of how the hand is currently used for this function or for similar functions. The hand-action and task primitive are compared on a measure-by-measure basis. If the hand action is deficient in any one measure for which the task primitive has a strong requirement, then the hand action can be selectively modified or a similar hand action chosen to correct exactly that deficiency. Through a process of successive refinement the design will converge on a satisfactory solution.

For each class and element of hand action there should be set of measures characterizing the action. For instance, for index MCP flexion there should be a description of its degrees of freedom, constraints, ability to coordinate with other joints, resolution, speed, and so on. For many hand actions the necessary measures are not available, or have been reported in a form not directly applicable to whole-hand input. In these cases, designers must make informed estimates, or perform experimental trials to determine appropriate values. Section 11 recommends experiments to measure hand-action capabilities. As this work is performed, a body of knowledge will be developed to quantify whole-hand input.

Task requirements and hand action capabilities do not always have one-to-one mappings. For instance, the hand degrees of freedom may map to a task function rather than to a task degree of freedom, e.g., using the pinkie may imply one mode of operation, while using the index finger, another. Some task requirements are rigid and the hand must accommodate, while others are more flexible and can be adjusted to accommodate to hand action capabilities.

Task characteristics and hand action capability measures may reflect quantities and qualities more than one way. Although represented as single values, some measures may apply to a vector of degrees of freedom; an abstracted task or hand feature; or a symbolic posture or gesture of the hand. For instance in the hand action capabilities, resolution may be used to describe the ability to bend the middle finger to 42.5°, the ability to hold the index and thumb .3 inches apart, the ability to move a fingertip in a circle, or the ability to separately sign the letters S and T of the Single Hand Manual Alphabet (Figure 8).

figs/hand-signs-S-T.eps

Figure 8: *The letters S and T of the Single Hand Manual Alphabet.*

Task characteristics and requirements

Quantitative measures

1. Degrees of freedom: the number of independent degrees of freedom. For instance, controlling a gauge is a one-degree-of-freedom task.

2. Task constraints: the position, motion, or temporal constraints imposed on the task. Task constraints fall into four categories:

- range of motion of the degrees of freedom,
- physical constraints (e.g., cramped cockpit, gloves, space suit),
- temporal constraints (e.g., feedback loop delay, equipment speed), and
- external forces (e.g., water viscosity, or g-forces)

Task constraints interact with many of the measures of hand actions and influence other measures of the task. For example the constraint that all actions must be performed wearing heavy gloves, limits range of motion, precision, and speed of hand actions. NASA and the military have studied glove-related constraints for tasks requiring spacesuits and chemical warfare protection (Durlach, 1989; Ervin, 1988). A temporal constraint, such as the delay in the feedback loop due to communication distances (Earth–Moon, for instance), affects the speed with which a sensory feedback task can be completed. This influences the optimum speed of hand motion, and perhaps steadiness requirements of the task and hand motion.

3. Coordination: the number of simultaneously controlled degrees of freedom a task requires. For instance, screwing a nut on a bolt initially requires six degrees of freedom to position and align the nut with the bolt. Once the nut is fitted, the constraints of the bolt reduce the task to one degree of freedom—the twist of the nut.

This strict description of coordination does not always indicate the difficulty of the task.

The method of performing the task is also important. For instance, the second stage of screwing a bolt on a nut is a one-degree-of-freedom task if done with a socket driver, but if physically performed with the fingertips, it requires the coordination of four or more degrees of freedom of the hand. How the degrees of freedom are related to each other also affects the difficulty. A six-dimensional position and orientation task is done easily with free-hand motion, but simultaneously controlling six unrelated scalar values can be difficult.

4. Resolution: how many values must be individually addressed. Expressed as the number of bits required to represent the achievable values. Resolution should be measured in both the *temporal* and *spatial* domains. Resolution is also a measure of the precision required for a task.

5. Speed: how quickly a value must be reached; how quickly a task must be repeated.

6. Repeatability: how accurately must a value or action be repeated. Repeatability can be expressed as percentage value. (Best expressed as percentage success, i.e., 95% indicates a high degree of repeatability.)

7. Steadiness: the length of time that a value must be maintained, or a pattern of motion repeated in the short term (minutes). Also, steadiness refers to the smoothness with which a sequence of values must be followed.

8. Endurance: the length of time that a task must be performed in the long-term (hours). Endurance is separate from steadiness because some tasks require long term endurance, but not short-term steadiness. Puppetry for film is one example. Performers do not need to maintain any one pose or motion for a long time, but they must be able to perform similar actions repeatedly over the course of a day-long session.

Qualitative measures

9. Expressiveness: the necessity of human qualities of emotion and indeterminacy, as in controlling a puppet. Expressiveness is difficult to quantify, however, it may be possible to characterize the expressive requirements of tasks in terms of speed, acceleration, jerk, and so on. The precise characterization of expressiveness requires further study.

10. Modality: range of function; how many modes are there to switch between; how diverse are the modes. For instance, fastening bolts may have two modes, aligning the bolt to the hole, and screwing it in. These two modes are mechanically and conceptually similar. If unscrewing the bolt was added, the modality would increase by a small amount. However, if the task included hand-signing a symbol for the correct size bolt, the modality of the task would significantly increase. Modality is the task counterpart to the *adaptability* measure of action capabilities.

11. Task analogy: the usability of existing skills for this task; the ability to frame the task in terms of other (more familiar) tasks. Task analogy gives a measure of how easy or difficult it may be to perform or learn a particular task based on prior knowledge or skill. For instance, in surgical simulation it may be preferable to emulate conventional methods, reducing the difficulty for the existing surgical community to use the simulator. This measure is the task counterpart to the *familiarity* measure of action capabilities.

Hand action capabilities

Quantitative measures

The first set are the quantitative measures of hand action capabilities. To help understand some of these measures, envision using a hand action such as the flex of a finger to change a setting of a numeric gauge. The measures can be viewed as the ability of the hand to control the setting with this action.

1. Degrees of freedom: the number of independent degrees of freedom. For instance, circling the tip of the index finger is a two-degrees-of-freedom task.

2. Hand constraints: internal constraints on the degrees of freedom of the hand. These are expressed as:

- range of motion of each degree of freedom,
- range of motion of combinations of degrees of freedom (such as the volume enclosed by the range of indextip motion),
- coupling between degrees of freedom,
- spatial interference between digits (fingers can't occupy the same space at the same time), and
- strength.

These can be influenced by a variety of factors, including anatomical structure, muscle strength, structural strength, and muscle innervation. For instance, the anatomical structure of the PIP joints constrains flex between approximately $0°$ to $120°$. Constraints also can be safety related. Certain limits on hand configuration and motion are important to avoid injury such as carpal tunnel syndrome.

3. Coordination: the number of simultaneously controlled degrees of freedom, and/or the ability to coordinate the motion with other degrees of freedom. For instance, circling a fingertip requires the coordination of two degrees of freedom, while flexing the index and middle fingers at the MP joint requires two one-degree-of-freedom coordinations. Some coordinations are easier than others. Forming a grip requires the coordination of many degrees of freedom of the hand; however, like many grips, it is in the repertoire of skilled motions already acquired by most people and does not require training or unusual skill. Another example is that the index MCP is easy to coordinate with the index DIP, middle MCP, and thumb MCP, but less easy with the pinkie MCP or the wrist roll. Constraints can reduce the difficulty of coordinated movement. Moving a finger in a straight line is easier when it is done against a straightedge. Precision and speed affect the difficulty of coordination; the slower or less precise the action needs to be, the less difficult it is.

4. Resolution: how many values can be individually addressed. Expressed as the least number of bits required to represent the hand action. Resolution should be measured in both the *temporal* and *spatial* domains. Resolution is a measure of precision; primarily a function of the physiological mechanisms controlling the degrees of freedom involved, and the skill of the person. For instance, informal observation suggests that the flexion of the index PIP has a resolution of about 8 bits, whereas flexion of the MP joint of the thumb is less precise with an approximate resolution between 6 and 7 bits.

5. Speed: how quickly a position can be reached; how quickly a pattern can be repeated. Precision and speed are related through Fitts Law (page 60). This must be taken into account when determining the difficulty of using a particular action for a particular task.

6. Repeatability: how accurately can a position, movement, or pattern be repeated. Repeatability can be expressed as percentage value. (Best expressed as percentage success, i.e., 95% indicates a high degree of repeatability.)

7. Steadiness: the length of time that a position can be held or a motion maintained in the short term (minutes). This is opposed to *endurance* which is a long term measure. Another measure of steadiness is how smoothly a sequence of values can be tracked. Some of the factors affecting steadiness are the likelihood for fatigue of a particular action and the physical constraints and supports in use. It is important to know what quality of the action enhances or hinders steadiness and which degree of freedom is most likely to fail first. For instance, a repeated finger gesture that includes the thumb held against the palm will fail sooner than the same finger motion with the thumb relaxed, because the thumb will tire against the palm. In other situations, this measure may indicate the addition of a physical support, such as an arm or palm rest for tasks that only require finger motion. The force required for the task also affects the steadiness of actions.

8. Endurance: the length of time that a usage pattern can be maintained in the long term (hours) without fatigue or injury. Like steadiness, endurance is affected by many factors. Endurance is separate from steadiness because some actions that are fatiguing in the short term, can be performed intermittently for hours.

Qualitative measures

Like task requirements, hand actions capabilities have qualities that are not as easy to quantify yet are useful characterizations.

9. Expressiveness: ability to convey human qualities of emotion and indeterminacy, as in an artistic performance. Expressiveness is difficult to quantify since it implies some symbolic interpretation by the receiver. Consider the motions of a musical conductor. Some of the motions, particularly those of the right hand, follow certain rules of interpretation having to do with tempo and beat, but other motions, usually made with the left hand, are up to interpretation and do not follow strict rules. The speed and jerk of left hand movement might indicate loudness in some instances, while the height of the hand indicates loudness in other instances. Some attempt can be made to define expressiveness in terms of measurable quantities such as speed of motion, acceleration, jerk, and extent of motion; but the mapping of these quantities is complex and open to interpretation. The quantification of expressiveness is an area which needs further study.

10. Adaptability: range of function; how easy it is to switch between hand actions, or to use a single hand action for more than one task. For instance, it is very easy to shift between a power grip and a precision grip. People instinctively perform this shift in the process of unscrewing a tight jar top. However, it may be harder to shift between flexing an index finger and twirling the tip of the pinkie. Much of this has to do with our familiarity with a particular action. The use of this measure is to help group adaptable actions to be used in different modes of the same task.

The other aspect of this measure, the adaptability of one action to several tasks, is useful if a limited set of actions are to be used across different modes of an application. Limiting the set of actions often simplifies the learning process, as exemplified by the Apple Macintosh interface. Most of the input is performed with the same point, click, and drag actions, although the effects change from mode to mode.

It may be possible to express adaptability in terms of numeric quantities such as degrees of freedom, coordination, or dynamic range. However, although a hand action with many degrees of freedom is more likely to be adaptable to several modes of use, other qualities of the action, too numerous to include in a definition of adaptability, may inhibit multi-modal use.

11. Familiarity: a measure of how familiar is a particular hand action. It has at least two sub-categories:

- similarity to existing skills, and
- similarity to everyday motions.

Familiarity yields a measure of how easy or difficult it may be to learn and use a particular hand action. For instance, because grips are used in everyday life, hand positions that mimic standard grips are simpler to learn than ones that are otherwise random. This measure is the hand action counterpart to the *task analogy* measure of task requirements.

6.5 Device capabilities

There are many different forms of capturing hand motion as discussed in Section 5.2. Each has advantages and disadvantages, and is appropriate for its own class of application. In this stage of the *design method*, a set of measures of device capabilities can be compared with task and hand-action measures to help choose the most appropriate technology for obtaining the whole-hand input data.

Device Capabilities

1. Degrees of freedom
2. Cross-coupling
3. Device constraints
4. Fidelity
5. Resolution
6. Steadiness
7. Reliability
8. Mass
9. Comfort
10. Convenience
11. Sampling rate
12. Computation required

1. Degrees of freedom: the number of degrees of freedom of the device. For instance, the Power Glove has 8, the DataGlove has 16, and the DHM has 20.

2. Cross-coupling: the extent to which the degrees of freedom are not independent; the nature of any cross-coupling between degrees of freedom of the device. For example, the DataGlove MP joint sensors have significant cross-coupling with neighboring MP joint sensors due to pulling of the glove fabric when the fingers are bent (Hong and Tan, 1989).

3. Device constraints: constraints on the degrees of freedom and use of the device. A degree-of-freedom constraint might be that the device will not measure hyperextension of the fingers (flex angles limited to positive values). A usage constraint might be that the device is delicate and cannot be handled roughly.

4. Fidelity: how accurately the device readings match the hand actions. This measure has two aspects. One is how well the device records the position of the hand—if a finger is bent at 22.3°, an accurate device will indicate 22.3° while an inaccurate device may indicate 20°, 23°, or 25°. Accuracy can be verified by direct measurement of the hand mechanically, or with X-ray, MRI, or other medical imaging techniques (Makower, Parnianpour, and Nordin, 1990; Wise et al., 1990). The accuracy of a device may also depend on the difficulty of calibration. A non-linear sensor with a known response curve can yield great fidelity with the proper software interpreting the non-linear response. Conversely, a sensor with a complex response curve that is difficult to calibrate cannot be used to measure hand position accurately (without a difficult calibration procedure).

The other aspect of fidelity is how well the device dynamically tracks the hand. Some of the factors that come into play here are shifting of sensors on the hand, sensor drift over time (in response to temperature, wear, and other factors), and consistency of response over the range of motion. For instance, not only do the VPL DataGlove sensors have nonlinear response curves to finger bends, but have different response curves when mostly flexed to when mostly straight (VPL Research, Inc., 1987). This makes calibration all the more difficult. In contrast, the DHM sensors have well known sinusoidal response and are adjusted so that the "linear" portion of the sinusoid is used to measure joint angles. For the most part, the sensor response is assumed to be linear, however, for supreme accuracy the known sinusoidal mapping could be used. Again, software can correct for these situations, if the response can be accurately mapped.

5. Resolution: How many useful bits of information are available from the device sensors. A device may have high resolution but low fidelity. For instance, the VPL DataGlove sensors have an eight-bit range, however their accuracy is not 0.5° as eight bits would indicate (120°/256, approx.) due to non-linear response in the sensors and slight shifts in the fit of the glove.

Resolution has two components, the resolution of the sensors themselves, and the effective resolution of the sensors in recording hand motion. The actual sensor resolution is dependent on the sensor technology and the analog-to-digital converters (A/D's) used. For instance the sensors on the Power Glove are capable of greater resolution than the two-bit A/D's allow. The effective resolution is a result of how much of the sensor range is used in the operation of the device. For instance, the VPL DataGlove has eight-bit A/D's, but to encompass a wide range of human hand sizes and hand ranges of motion, the average hand flexes the sensor through only fifty to seventy percent of its range. This reduces the effective resolution to six or seven bits. The DHM has similar constraints, however twelve-bit A/D's keep the effective resolution well above eight bits. Kramer's CyberGlove avoids this problem with dynamically tunable circuitry, so that in the calibration stage, the A/D inputs are scaled to encompass exactly the range of motion of the fingers.

For some applications, resolution is more important than fidelity. In the case of controlling the joints of two simulated robot arms in a grasping task (page 170), the resolution in the finger joint sensors seems to be more important than fidelity to the finger position. The main reasons for this seem to be that the finger-to-robot mapping does not have a 1:1 ratio so the interface needs a certain amount of practice before it can be mastered. The user eventually learns the mapping from hand-joint space to robot-joint space. Changing the fidelity of the finger-to-sensor mapping would only affect the intuitiveness of the control, changing the learning curve. However, changing the resolution of the control would affect the control task in a way that would affect task performance regardless of a user's practice time.

6. Steadiness: Steadiness has short-term (millisecond) and medium-term (minute) aspects. On the short-term side, steadiness refers to how much noise there is in the system (sensors plus electronics) or signal-to-noise ratio of signals. Steadiness affects both *resolution* and *fidelity*. A device that has high resolution in its sensing capabilities may also have a high rate of error. The Polhemus tracker is a good example of a device exhibiting this characteristic. It can track a location within a few millimeters, however it is very susceptible to electronic interference and the location values are relatively noisy. As another example, the Exos DHM sensors use 12-bit A/D's, but the least significant 2-3 bits are lost in sensor and electronic noise. On the medium-term side, steadiness is affected by sensor drift over time. Some device sensors exhibit drift in response to temperature or other environmental changes. Some devices shift on the hand during use, changing the sensor response.

7. Reliability: mean-time between failures under different conditions of use (this could be considered the long-term aspect of *steadiness*). For instance, the Power Glove is extremely durable, whereas the DHM can fail if used improperly or abusively.

8. Mass: the weight and inertia of the device. Mass affects user fatigue and the dynamic performance of the hand. For instance the DHM is more massive than the DataGlove and gives the hand unnatural inertia. For this reason, the DataGlove may be less tiring to use than the DHM. The DHM's added inertia makes the hand harder to wave around in the air than the DataGlove, reducing the acceleration of the motions that can be performed with the device.

9. Comfort: how comfortable the device is to wear. This is a subjective measure based on samples of users.

10. Convenience: how easy the device is to attach, put on, engage, or remove. This quality can be quantified by the length of time it takes to put on and remove a device, and the time it takes to calibrate it for normal use.

11. Sampling rate: how frequently the device reads and reports hand actions.

12. Computation required: how much computation is required to convert raw data into a useful form.

6.6 Use of the design method

The *design method* is intended to be used as an iterative process in which the designer makes initial design decisions as to the best input method for a designated task, and then refines those decisions with the information in the evaluation guide. The iterative design process is a natural approach to a problem involving as many variables as does that of human-computer interaction. The *design method for whole-hand input* is structured to clarify and ease the process, and to reduce the number of iterations.

It should not be expected that the *design method* itself will answer the whole-hand input design problem. Hard and fast rules for interface design are difficult to come by. Frederic P. Brooks, Jr. (1977) writes,

> *The architect Christopher Alexander in his* Notes on the Synthesis of Form *(1964), makes the penetrating observation that the only way to achieve good fit between any design and its requirements is to find misfits and remove them; there is no direct way to derive form from requirement. Good fit is the absence of all possible misfits. This he supports with convincing arguments.*
>
> *This I find to be an overarching lesson from all our graphics system design work. We observe that we have not found a direct design procedure for the man-machine interface; Alexander shows that we never shall. Principles we have found; we shall find more; and these will guide design. Satisfactory man-machine systems, however, will always be the product of iterative design in which the misfits are painstakingly removed. I think the only effective design methodologies will be those built around this iterative approach.*

Thus, the *design method* offers principles to guide whole-hand input design, and provides a degree of discipline to the iterations of the design process. The steps that a whole-hand input designer takes are as follows:

1. Answer the questions posed in Section 6.2 as to the appropriateness of the task to whole-hand input. This involves analyzing the task and hypothesizing various forms of using whole-hand input in the application.

2. Examine and hypothesize styles of interaction for the task based on the *taxonomy of whole-hand input.*

3. Divide the task into task primitives and analyze their characteristics and requirements, determining values for each of the measures in the evaluation guide.

   The evaluation guide lists the relevant categories to be analyzed. Previously analyzed task primitives may be similar, and the designer should use them as examples in the process. The designer should keep results of previous whole-hand input designs, building a knowledge base of whole-hand input design into the evaluation guide.

4. Find or devise appropriate whole-hand input methods for the task primitives by matching task characteristics and requirements with the existing knowledge base of hand actions in the taxonomy and evaluation guide. Preliminary choices of hand actions may be based on literature reviews, personal experience, or direct observation of current use of the hand in the task or in similar tasks.

   Task characteristics and requirements are matched with whole-hand input methods through the measures of hand action capabilities. The list of task requirements can be used as a template to overlay the list of hand action capabilities. This is an iterative process involving modification of hand actions (or even task requirements) along specific attributes found to be deficient in measure-to-measure comparisons.

   Interpretation of hand actions also affects which control methods are selected from the evaluation guide. Many hand actions can have functionally different characteristics depending on whether they are interpreted as direct, mapped, or symbolic controls. Some that are appropriate for one style of input may not be appropriate for another. The designer must take into account the nature of the task that might recommend one or more of these interpretations.

   This dissertation provides a framework to begin the development of a knowledge base of quantitative and qualitative measures of whole-hand actions. There are an unlimited number of possible hand actions; not all can be listed in the evaluation guide. One method of dealing with this problem is to classify hand actions so that representatives of each class can be used to guide the design process. The categories in the evaluation guide help to classify hand actions for this purpose. The available knowledge base will grow as designers develop new methods of whole-hand input and enter them into an evaluation guide.

5. Compare characteristics of the chosen methods for whole-hand input control (along

with the task characteristics and requirements) with device capabilities to choose the appropriate device(s) for the tasks.

6. Implement a simulation of the tasks and the chosen whole-hand input methods in a testbed (such as described in Section 8). This may entail developing new code, assembling pieces of existing code used in previous simulations and methods, or using existing code outright.

7. Test methods in the task simulation with the chosen device(s).

   For each test, analyze the entries in the evaluation guide as to their importance to the success or failure of the performance of the task.

8. Refine the task interaction by searching the evaluation guide for similar methods that preserve the elements that contribute to performance success and change those elements responsible for performance failure.

   For instance, if the degrees of freedom and coordination are well matched, but not the resolution, then a different method can be chosen that has the same degrees of freedom and coordination, but different resolution.

9. Repeat the process from step 4 (or before, if it is found that the task itself has been improperly designed or can be improved, or if another style of interaction would be more appropriate) until the task can be accomplished within the specified requirements.

7 Evaluations of Whole-hand Input

This section reports on three prototypical user trials that were performed to validate the principles of the whole-hand input design method described in the previous section, and to contrast task performance using whole-hand input versus more conventional devices. Three tasks were chosen to illustrate the range of performance of whole-hand input versus conventional input devices; one in which the design method predicts superior performance with whole-hand input, one in which whole-hand input is predicted to perform similarly to conventional input, and one in which conventional input should outperform whole-hand input. They are presented in this order.

In each of the evaluations, the VPL DataGlove was used as the whole-hand input device and an eight-knob box along with a 32-button box used for conventional input. The tasks were simulated on an HP9000-series 835 Turbo-SRX graphics workstation using the bolio interactive graphical simulation platform (see Section 8). Of the three available whole-hand input devices, the VPL DataGlove, the Exos DHM, and the Mattel Power Glove, the DataGlove was chosen over the DHM and the Power Glove for ease of getting on and off the hand, calibration ease, and comfort.

Compared to the DataGlove, the DHM is difficult to don and doff, and requires more careful calibration procedures.[1] The Power Glove does not have these problems, however it lacks the necessary precision and degrees of freedom for the tasks. The improved resolution and sensor independence of the DHM over the DataGlove was not enough of an advantage to warrant its use in this case were the whole-hand input device was put on and removed several times over the course of a subject's session, fast calibration was important to save time, and the limited resolution of the DataGlove was adequate for novice use.

The knob (or dial) box and button box were chosen as representative of commonly used devices in interactive computer graphic simulations and applications, and because they are supported by Starbase (the HP workstation graphics system). Joysticks were considered, but are not supported by Starbase. Although mice are often used with graphic interfaces they are less frequently used in 3D interactive computer graphic applications than dials

[1] The DHM can be configured with fewer sensors for specific tasks, thus easier to put on the hand. In the tests done here, a thumb plus two fingered DHM with only MCP and PIP sensors would have sufficed and been easier to put on subjects than the fully arrayed 20-degree-of-freedom DHM that was available. Fewer sensors also eases the calibration process.

and buttons. In addition, a mouse lacks the degrees of freedom needed for the tasks.

None of the subjects participating in the trials had prior experience using whole-hand input devices, or using conventional devices in an interactive computer graphic environment. The order of the evaluations was maintained for each of the subjects—the order presented here—but within each task the order of glove versus conventional devices was selected randomly.

The three tasks were *walking*, *object orientation*, and *path following*. They are based on sub-tasks within the six-legged walker application described in Section 10.1. In each of the following descriptions, the task is described briefly along with the experimental method. Then a single pass of the whole-hand input design method is applied to the task (with the exception of the device evaluation) to both illustrate the use of the method and to predict the effectiveness of the application of whole-hand input to the task as opposed to conventional inputs. Each section is concluded with the experimental results and discussion.

7.1 Walking task

The goal of this evaluation was to test a task for which the design method predicts superiority of whole-hand input over conventional devices. The task chosen was low-level control of walking of the six-legged walker described in Section 10.1 (see page 168). Ten subjects participated in this evaluation.

The task and experimental method

The six-legged walker has a low-level locomotion control in which the user controls the individual joints of the forelegs and the middle and rear sets of legs follow in a tripod gait (alternating feet on the ground forming a tripod of support under the center of gravity). The whole-hand input control for the task uses the index and middle fingers to mimic the walking motion of the legs. The index MCP controls the left hip joint while the index PIP controls the left knee joint. The middle finger controls the right leg similarly. Conventional control is with four dials, one for each of the leg joints. The leg joints have fixed rotational limits beyond which the dials (and the finger controls) have no effect. Turning the walker figs/walk-task.eps

Figure 9: *The walking task* The six-legged walker appears on the left. The whole-hand and conventional control inputs are illustrated on the right. The user's index and middle fingers control (joint-to-joint) the hips and knees of the walker's forelegs. The middle and rear legs follow in a tripod gait. Alternately, four dials control the four joint rotations. (Other controls for the walker, including use of the two arms, are discussed in Section 10.1.)

was disabled to simplify the task.

Subjects were asked to "walk" the walker for a length of time, continually trying to improve their technique and achieve the greatest average walking speed they could manage. Throughout the trial, the author coached the subjects in "walking" technique. There was no distinct training phase for the task since subjects continually improved their technique. At some point their performance leveled out. Although they still made occasional mistakes, became fatigued, and experimented with variations in technique, their average speed remained more or less constant.[2] At that point the task was ended, and their best time automatically recorded. This usually occurred after ten or fifteen minutes of performing the task. This was done once using whole hand-input, and once using the dials.

The view camera tracked the walker, so it was never off-screen. Just before the walker

[2]People with more whole-hand input experience were informally tested and performed significantly better at this task, indicating that the performance plateau experienced by the novice test subjects would eventually be surpassed in a second wave of improvement.

got too far from the camera to be seen clearly (about fifty seconds with whole-hand input and two minutes with conventional input) the walker was reset to the center of the world, and the subject continued. The reset operation took less than a second and was a minor interruption in the flow of the task. (Having the camera position follow the walker, keeping the walker centered in the view was tried and abandoned because subjects were less aware of their progress, and subsequently less motivated, than when the camera position was fixed and rotated to follow the walker as it walked away.)

Performance was evaluated in terms of the best average speed of the walker over a thirty-second period. The simulation update rate was about ten hertz, and every three frames (or approximately three times a second) the position of the walker automatically was recorded into a data file. Recording took place from the beginning to the end of each trial, including the time the subject was learning the task, experimenting with alternate techniques for improving performance, and concentrating on walking quickly.

To properly evaluate the data, the fastest average time over a thirty-second period was culled from the readings. To do this, the data file was processed to remove the effect of the position resets. At each reset point, the position just prior to reset was added to subsequent position data. The effect was as if the walker moved continuously in one direction. Then a linear regression analysis was run on every consecutive thirty seconds of data (about 1700 regressions for ten minutes of data using a moving window of 90 samples). The best of these was taken as the metric of the task.

Application of the design method

Appropriateness of whole-hand input:

Is the use of whole-hand input appropriate and beneficial to the application task?

Naturalness

Are pre-acquired sensorimotor skills useful for controlling the task?

> Possibly. For most people, walking is a natural function. It may be possible for people to translate the sensorimotor skills of leg motion to the fingers.

Are existing hand signs skills useful for controlling the task?

No. The task is a continuous process with no existing convention for hand signing (as there is for traffic control, for instance).

Is the absence of an intermediary device useful for controlling the task?

Unknown. An intermediary device may provide useful constraints on finger motion, but most likely only if the device was specially designed for this task.

Can task control map well to hand actions?

Yes. There are various ways that leg joints or foot position can be mapped to hand actions such as leg joints to finger joints, fingertip position to foot position. (Other mappings suggest themselves if higher-level leg control is possible, such as MCP flex to gait cycle, or tapping fingers to indicate gait pattern and speed.)

Adaptability

Are diverse modes of control used in the task?

Not for this specific subtask. Only one mode of control exists for low-level control of the walker. However, since this is a sub-task of a larger application with many modes of control, the answer for the whole application is yes.

Is it important to be able to switch between modes of control rapidly, and smoothly (i.e., with a minimum of distraction from the task)?

Not for this specific subtask. See above. For the whole application however, it is important to switch smoothly between the modes for efficient execution of the task.

Coordination

Does the task require the coordination of many degrees of freedom?

Yes. There are four degrees of freedom that require careful coordination to move the walker forward. Poor coordination will position the legs improperly or sequence the motions improperly, retarding progress or even causing the walker to walk backwards.

Real-time control

How important to the task is continuous monitoring?

Important. There is no automatic control system at this level of control of the walker. Without continuous input the walker does not move.

How important to the task is rapid user response?

Minor importance. At any instant the walker is dynamically stable precluding the need for rapid reactions to perturbations in the walking system and there are no sudden environmental events to react to. If the task were performed in an environment that required the walker to avoid moving objects or react to sudden occurrences, then rapid user response would be important.

Taxonomy – input style:
What style of whole-hand input is intended and/or seems to be best suited for this task?

Continuous/Direct	Continuous/Mapped	Continuous/Symbolic
Discrete/Direct	Discrete/Mapped	Discrete/Symbolic

This task seems best suited for a Continuous/Direct style of interaction based on the observations above and the desire for a low level of control over the walker's legs.

Evaluation guide

Task characteristics and requirements

The task has only one functional unit: coordinating the four degrees of freedom of the legs to move the walker forward. This is the "task" analyzed with the evaluation guide.

Degrees of freedom: The task has four degrees of freedom: flex angles of the left hip, left knee, right hip, and right knee.

Task constraints: All four joints have constraints of maximum and minimum flexion: $0°$ to $120°$ for the knees, $-70°$ to $70°$ for the hips. There are no physical constraints on the user, temporal constraints beyond those imposed by the graphics update rate, or external forces.

Coordination: All four degrees of freedom must be coordinated. However, the degrees of freedom can be operated independently at the expense of the speed of operation.

Resolution: Only moderate resolution is required of the task, perhaps five to six bits minimally. Exact foot placement is not necessary, but to make progress, the feet must be planted ahead of the "hip joint"—the further forward the better. The resolution of joint angle control affects the ability to control the speed of progress, foot placement, and ability to coordinate the gait. Too little resolution can make the task difficult to impossible.

Speed: The task has no speed requirements except as relates to how rapidly the task is to be performed, i.e., the task does not inherently require fast performance. It should be noted that the update rate of the simulation for this task is approximately 10 hertz.

Repeatability: The repeatability requirements of the task are not stringent. It is not important that the legs always bend at the same angle at the same part of the gait, however it is important that the legs touch the ground in the correct sequence, and that the lower-leg not on the ground is tucked out of the way during the forward swing phase of the gait.

Steadiness: Steadiness is not important (i.e. less than 0.5 seconds) while walking, but if holding the walker at a certain location is required then steadiness is important for as long as the walker is required to maintain a position. For the purposes of the experiment, steadiness is less than 0.5 seconds.

Endurance: The task must be performed for at least ten to fifteen minutes for the purposes of the experiment.

Expressiveness: There are no requirements of expressiveness for this task.

Modality: This task has only one mode. However, if part of a larger application, there must be a way to shift out of the one mode.

Task analogy: The task has many of the kinematic aspects of human walking. Conventional methods for interactive control of robot limbs use buttons or joysticks to control individual degrees of freedom or the end-effector of the limb.

Hand action capabilities

In practice, the designer selects potential hand actions that appear be most effective for the task, and references those capabilities which have been previously evaluated and records new evaluations for those which have not. Then, these are compared to the task requirements, modifying the hand actions or selecting new hand actions until one is found that satisfies the task requirements. The resulting method is tested with the application and, if necessary, refined with further iterations within the design method. In the interests of brevity, only the hand action capabilities of the final whole-hand input method, *flexion of the index and middle finger MCP and PIP joints*, are presented here with brief explanations of why the particular method was selected.

Some of the measures have been estimated because clinical studies that address the particular measure could not be found in a literature search or do not exist. Following each entry in the hand action capabilities is a short commentary and comparison to the corresponding task requirement.

Degrees of freedom: This hand action has four degrees of freedom: the index MCP flex, index PIP, middle MCP flex, middle PIP. *The four degrees of freedom equal the four degrees of freedom of the task.*

Hand constraints: The American Academy of Orthopaedic Surgeons (1988) places average MCP joint limits at 90° flexion and 45° extension, and average PIP joint limits at 100° flexion and 0° extension, based on a neutral position of fingers parallel to the palm.[3] Informal observations by the author estimate that due to the ligament structure of the hand, the index and middle MCP joints must be flexed within approximately 45° to 55° of each other.[4] *The finger-flex constraints are similar in geometry to the task's leg joint constraints although different in value. The selection*

[3]Studies by An et al. (1979) found MCP joint limits of +85° and −22° and PIP joint limits of +102° and −11° based a relaxed joint position.

[4]This estimate is subject to formal experimentation. A review of the literature did not indicate that this measure of finger correlation has been the subject of any formal studies.

of finger flex over other hand actions allows the natural constraints of the fingers to assert the constraints of the legs. The correlation will limit opposing motion of the legs, but should not overly constrain the ability to walk.

Coordination: Although the four joints are generally independent, many common tasks are performed with coordinations of the index and middle fingers. Consequently the hand is predisposed to the coordination of these joints. *This compares favorably with the task requirement that the degrees of freedom be coordinated. The thumb and index are arguably more coordinated in general than the index and middle fingers, however, the latter have a better mapping to the task's walking action.*

Resolution: Unknown. The index and middle finger are arguably the most precise of the fingers, however a review of the literature has not revealed a study indicating the precision of those actual joints.[5] *Although the actual resolution is unknown, these joints are among the best available choice in terms of the task resolution requirements (the others being on the thumb and pinkie).*

Speed: Unknown.[6] Informal observations put the speed of the joints at approximately 0.5 seconds to achieve any specific angle, and skilled users can repeat the "walking" pattern at approximately 1.0 hertz. *Although there are no specific speed requirements for success in the task, the index and middle fingers can move fast enough given the update rate of 10 hertz and have more than adequate capacity for rapid task performance.*

Repeatability: Unknown. This measure is dependent on the nature of the exact motion that needs to be performed. For most motions it is expected to be high, given the dexterity of the index and middle fingers. *Informal observations put the repeatability of the hand action well within the task requirements.*

Steadiness: The index and middle fingers are quite strong and can be held in position for several minutes before fatigue sets in. *This is well within the steadiness requirements of the task.*

[5] Mesplay and Childress (1988) found MCP joints to have an information transfer rate of 4.14 ($\pm$0.89) bits/second in a pursuit tracking task; better that the wrist and elbow joints. This suggests the superiority of precise control from the MCP joints over the whole hand itself, but the study did not comment on the resolution of the joint itself.

[6] Mesplay and Childress (1988) report maximum MCP angular accelerations of 11300 rad/sec^2.

Endurance: Unknown. Informal observations indicate that this motion can be performed with minimal interruption for up to ten minutes, and for much longer periods of time if there are opportunities to rest. *This is approximately matched to the endurance requirements of the task.*

Expressiveness: These two fingers have a great deal of dexterity and are used expressively in a wide variety of functions including drawing, painting, playing musical instruments, and gesturing. *The task requires no expressive capabilities.*

Adaptability: These fingers are used for a variety of everyday purposes and so can be rated with a high level of adaptability. *The task has only one mode. Adaptability is not relevant.*

Familiarity: The index and middle fingers are used in many everyday functions and so their use is highly familiar. The particular coordination of "walking" with them also is a familiar action to most people. *The high task analogy and familiar movement give this hand action a highly favorable recommendation for use for this task. Other hand actions fare less well in this category, given the nature of the task.*

Commentary

The design method indicates that using the whole-hand input method chosen should allow users successful and efficient control of the task. The number of degrees of freedom that need to be coordinated and their natural mapping to the hand, indicate that the use of whole-hand input is appropriate for this task. The task requires four degrees of freedom whose coordination is crucial. The hand action chosen has the requisite degrees of freedom and features a high degree of dexterity and coordination when mapped to the task. There are no conflicts between the task requirements and the hand action capabilities of the whole-hand input technique chosen, thus "walking with the fingers" is indicated as a good whole-hand input strategy for this task.

Comparatively, the dials allow control of the requisite number of degrees of freedom, but do not aid the task of coordination. The operator is reduced to manipulating one, or two degrees of freedom at a time, reducing task performance. The dials allow greater precision than the DataGlove, however only moderate precision is required of the task. The endurance level of the hand motion is about equal to the duration of the task, and in fact, partly determines the length of the trials. Consequently, it is predicted the whole-hand input will allow better performance for this task than will conventional input devices.

Test results and discussion

Overall, subjects found walking with the glove very natural; so much so that they played with the walker, making it walk forwards, backwards, dance, and run. It took, on average, three to five minutes to become comfortable with the technique, although some mastered it almost immediately. With the dials, most subjects picked up the idea quickly and reached a steady-state performance level within two minutes. Beyond that there was little improvement with the dials. Experienced DataGlove users who tried the system (but were not included in the statistical analysis) performed equally well with the dials as the novice users, and performed much better with the glove then the average novice. This suggests that over longer periods of time performance with the glove will improve, while dial performance will not.

One of the problems with the dials was the delays incurred in moving the hand from one dial to another. In addition, subjects found it difficult to develop a successful pattern of dial sequencing and often found themselves reaching for the wrong dial. When allowed to use two hands, subjects found the coordination of the two hands on the dials just as difficult and performed no better than with one hand. Subjects also reported that the dials took significantly more concentration to operate smoothly (high cognitive load) than the glove.

Statistical results of this task evaluation are shown in Figure 10. One can observe that glove walking speeds are well above (with two exceptions) dials walking speeds, and that the variation of glove speeds in comparison with the variation of dial speeds indicate that there is a narrow band of performance available with the dials, while individual skill with the glove interface strongly affects performance. Performance with the glove seems to be, on average, three times better than that with the dials, and the best of the novices, who was also the fastest with the dials, was almost five times faster with the glove than with the dials. Experienced whole-hand input users can move the walker up to ten times faster with the glove than with the dials.

To determine the confidence with which the data represents all potential novice users, a t-Test using a pooled estimate of σ^2 was performed. The null hypothesis was that the means of the two variables are statistically equal. The hypothesis was rejected with a certainty of over 95% (Figure 10) indicating that the data does indeed represent potential users.

figs/walk-results.eps

Figure 10: *Results of walking task* Units are in distance/time. The data has been scaled to be in (approximately) walker body-lengths per second. The dashed line shows the line of equal performance, where the subject would have performed equally well with either device. Points above the line represent subjects that do better with the glove than with the dials.

7.2 Object orientation task

The goal of this evaluation was to test a task for which the design method predicts similar performance with dials and glove. The task chosen was high-level control of the orientation of an object in the grasp of the six-legged walker described in Section 10.1 (see page 175). Twelve subjects participated in this evaluation.

The task and experimental method

figs/rot-task.eps

Figure 11: *The orientation task* On the left is the subject's view of the orientation task from just behind the walker's head. On the right are the whole-hand input controls and conventional controls used for the task. The control stick held in the walker's arms and controlled by the user to match the reference stick. Note that the reference stick is kept centered on the controlled stick.

The six-legged walker has high-level controls for position and orientation of an object held in its grasp. The user specifies rotational and translational motions of the object and an inverse kinematic function is used to rotate the arm joints to achieve the object motion specified. The whole-hand input control for the task uses the rotation and translation of the hand (recorded by the DataGlove's Polhemus in the experiment) to control the rotation and translation of the object. A fist posture acts as a button to enable rotations. As long as the user holds a fist, the object will follow the hand's motions relative to the hand's position and orientation at the point the fist posture was entered. The user can "regrasp" the object by opening the hand, reorienting, and forming a fist again. Conventional input uses six dials to control the six rotational and translational degrees of freedom of the object.

Subjects were asked to match the rotation of two rigid rods, one held in the walker's grasp and called the "control stick," and the other rotationally static, called the "reference stick." To isolate the rotation aspect of the task, the reference stick followed the control stick's translations maintaining a common center with the control stick. Rotation about the axis of the control stick was ignored in the orientation match, as the subject had little control over that degree of freedom.

When the control stick was rotated to within three degrees of the reference stick and held for at least two update frames (so the subject could not simply "pass through" the goal orientation), the reference stick's color blinked several times, after which the reference stick automatically rotated to a new orientation, randomly picked from a table of thirty random orientations. A table of random orientations was used rather than randomly calculating a new orientation, so that subjects all were performing the match against a similar set of orientations. Some of the orientations were harder to achieve than others, so having a fixed set of orientations reduced the probability that one subject would have more "hard" or "easy" orientations than another. Subjects were given a maximum of sixty seconds to achieve any one match, after which a new reference orientation was selected automatically.

Each subject had approximately five minutes of practice with each of the devices (longer if they had particular troubles learning the task, and shorter if they mastered it easily). After practicing with both DataGlove and dials they were timed to see how many matches they could make in a five minute time period with each method of control. The time for each match was recorded automatically, and the average matches per second was calculated for successful matches (those under sixty seconds) and used as the metric for the experiment.

The task was made more difficult by several factors. One was noise in the Polhemus readings. This made it difficult to hold the object steady near the goal orientation. The dials did not exhibit this problem and gained an advantage in this regard. Another difficulty was in the nature of the task. As described in Section 10.1 the user does not rotate the object directly, but sets goal rotations (usually of small variance from the current orientation), which the arms then attempt to reach using an inverse kinematics algorithm. If the goal can not be reached given the initial state of the arms, the object does not move. This occurred more with the glove since high-speed rotations were possible, increasing distances to goal rotations. The dials were less prone to this since rotations were composed of smaller increments.

In addition, the kinematics of the arms occasionally caused rotations slightly deviated from the subject's indented goal, especially at singularities of the inverse kinematics. If the goal rotation was near a singularity in the arm system, then the subject had a particularly difficult time. Subjects reported that certain orientations were hard to "read," and that although they used parallax, occlusion, and perspective as alternate depth cues, they still had difficulties. The common strategy to overcome this problem was to decompose the degrees of freedom, rotating in one dimension at a time. In these situations the dials had the advantage since the degrees of freedom are naturally decoupled, whereas with the glove they can be decoupled only to the extent of the motor-skills of the subject. Conversely, the glove input had the advantage when the goal orientation was clear and a simple rotation of the hand could suffice.

Application of the design method

Appropriateness of whole-hand input:
Is the use of whole-hand input appropriate and beneficial to the application task?

Naturalness

Are pre-acquired sensorimotor skills useful for controlling the task?

> Yes. Most people have developed the ability to orient objects with the hand, a task used in everyday life.

Are existing hand signs skills useful for controlling the task?

> Possibly. There are hand signs that commonly can be accepted as indications of translation and rotation. For example, pointing in the direction of translation or circling the hand in the direction of rotation.

Is the absence of an intermediary device useful for controlling the task?

Unknown. An intermediary device may provide useful constraints to the task.

Can task control map well to hand actions?

Yes. The hand can be translated and rotated quite easily to mimic the translation and rotation of the object.

Adaptability

Are diverse modes of control used in the task?

Not for this specific subtask. Only one mode of control exists for high-level manipulation of the walker's arms. However, since this is a sub-task of a larger application with many modes of control, the answer for the whole application is yes.

Is it important to be able to switch between modes of control rapidly, and smoothly (i.e., with a minimum of distraction from the task)?

Not for this specific subtask. See above. For the whole application however, it is important to switch smoothly between the modes for efficient execution of the task.

Coordination

Does the task require the coordination of many degrees of freedom?

Perhaps. Basically there are three degrees of freedom to be controlled (the translations play a minor or no role). Coordination of the three degrees of freedom may speed execution of the task in many circumstances, but the task can be accomplished with decoupled degrees of freedom, and in some situations is easier with the degrees of freedom decoupled.

Real-time control

How important to the task is continuous monitoring?

Important. There is no automatic, supervisory control of the orientation of an object in the walker's grasp. Without continuous input the object does not move.

How important to the task is rapid user response?

Minor importance. There is no movement without input, and there are no sudden environmental events to react to. If the task required the orientation to be done very quickly, or there were dynamic constraints on the object (such as rotational inertia that would require compensation from the operator), or external occurrences (such as aligning the object with a moving target), then rapid user response would be important.

Taxonomy – input style:
What style of whole-hand input is intended and/or seems to be best suited for this task?

Continuous/Direct	Continuous/Mapped	Continuous/Symbolic
Discrete/Direct	Discrete/Mapped	Discrete/Symbolic

Based on the observations above, this task well suited for a Continuous/Direct or Continuous/Mapped style of interaction. The task requires continuous monitoring of the orientation of the control stick and so a continuous style of hand action. On the application side, input from the hand is mapped to rotations of the stick. In the sense that hand orientations conceptually correspond to control stick orientations a direct interpretation is used. However, computationally, the hand motions are mapped to robot arm motions through an inverse kinematic routine, and are represented by a Continuous/Mapped style of input.

Evaluation guide

Task characteristics and requirements

The task (as presented) has two functional units: orientation of the control stick and engagement (or disengagement) of the hand to the task. The most important function is the orientation of the control stick. This is the one analyzed in the following evaluations.

The other function has a parallel analysis which has been omitted for the purposes of this text.

Degrees of freedom: The task has six degrees of freedom: three translational and three rotational. However, the three translational play little or no role in the performance of the task.

Task constraints: The motion of the object is constrained to follow the kinematics of the robot arms. Otherwise there are no constraints.

Coordination: Coordination of the six degrees of freedom is not essential to the success of the task. The degrees of freedom can be decoupled at the expense of the efficiency of task performance.

Resolution: The sticks must be within 3° of each other to match. This requires a resolution of 360°/3°, or 7 bits.

Speed: The task has no speed requirements except as relates to how rapidly the task is to be performed, i.e., the task does not inherently require fast performance. It should be noted that the update rate of the simulation for this task is approximately 4 hertz.

Repeatability: The repeatability of the task is determined by the tolerance needed to match the sticks. At 3° in 360° the repeatability is approximately 99%, i.e., any orientation must be achievable within 99% of it's true value.

Steadiness: The match must be held for at least 2 frame times, or 0.5 seconds.

Endurance: For the purposes of the experiment, the task must be performed for at least five minutes without resting.

Expressiveness: There are no requirements of expressiveness for this task.

Modality: This task has only one mode. However, if part of a larger application, there must be a way to shift out of the one mode.

Task analogy: The task is similar to everyday rotation of objects, but differs in the quality or absence of sensory feedback, and in the constraints imposed by the walker's arm kinematics. Conventional interactive manipulation of objects in robotic grasps use joysticks or buttons to control robotic joints or control the end-effector using euler angles or orthogonal axes. Conventional methods for interactive computer graphics object orientation use buttons, dials, or mice to control the degrees of freedom in a cartesian coordinate system.

Hand action capabilities

As was done with the previous experiment (see page 108), only the evaluations for the hand action chosen at the end of the iteration process, *hand orientation*, are presented here. Again, informal observations are used for some of the measures in the absence of clinical data. Following each entry in the hand action capabilities is a short comparison to the corresponding task requirement.

Degrees of freedom: Hand orientation and position has six degrees of freedom: three rotational and three translational. *These six exactly match the six of the task. No other hand action has this kinematic correspondence.*

Hand constraints: (Translational hand constraints are not important for this task, so only rotations that preserve position are considered here.) The hand cannot be rotated through 360° in any one axis without rotations in the other axes to overcome the kinematic constraints of the wrist, elbow, and shoulder. The American Academy of Orthopaedic Surgeons (1988) puts the average range of wrist flexion/extension at 130°–150°, ulnar/radial deviation at 45°-60°, and, forearm pronation/supination (wrist rotation) at 140°-160°. These rotations tend to incur very little translational movement of the hand. To the wrist range of rotations can be added coordinated shoulder and elbow rotations that preserve the position of the hand. These can add an estimated 60° to 120° to the range of hand orientation. *In the experiment, it does not matter which end of the control stick is matched to which end of the reference stick. Therefore, the subject never has to rotate the control stick more than 180° in any one direction, and on average has to perform a 90° rotation to achieve a match. Thus, the average match will be within the hand's rotational limits. In cases where the rotation is great, or the kinematics of the walker arms are such that it is better to rotate through the larger angle, the limits on hand orientations can be overcome by "releasing" and "regrasping" when the hand's rotation limit is reached.*

Coordination: The six degrees of freedom of the hand can be highly coordinated as a consequence of everyday use. It is difficult to decouple the degrees of freedom, however. *This will be an advantage at those times when the user wants to coordinate the degrees of freedom, and a hindrance when the user wants to decouple the degrees of freedom. Other hand actions, such as the flexion of each of three fingers could be used to decouple the degrees of freedom, either as switches in conjunction with the hand orientation (flexing a finger turns on or off a degree of freedom), or by mapping a rotational degree of freedom to each of the finger joints. In a more intensive interface design process, further iterations of the design method (if it is determined that decoupling is a necessity) should examine this hand action.*

Resolution: Unknown. The wrist, elbow, and shoulder determine the orientation of the hand (relative to the body) and thus determine the resolution with which the hand can be oriented. *Experience with whole-hand input and three-space trackers indicates that hand orientation has enough resolution for the task. In fact, the 3° tolerance for the task was based on preliminary trials taking into account hand and Polhemus tracker accuracies.*

Speed: Unknown. Informal observations put the speed of a rapid wrist motion at 300–400 degrees/second. *Rotational speeds in the task are limited by the 5 hertz update rate. Experience with similar interactive computer graphic applications indicates that the rotational speeds of the hand are more than adequate for the task.*

Repeatability: Unknown. *Informal observations and experiments by Drucker (1990) put the ability of the hand to perform this kind of task (and thus repeatability of hand orientation) within the requirements of the experiment.*

Steadiness: Informal observations indicate that the hand can be held unsupported in place for 30 to 60 seconds before fatigue causes unsteadiness. If the forearm is supported the time period is greater. *This is well within the 0.5 seconds required for the task.*

Endurance: Informal observations indicate that positioning of the hand can be performed on the order of hours given occasional periods of rest. This is probably due to the extensive use of the hand for everyday tasks. *This is well within the ten minutes required for the practice and trial runs of the experiment.*

Expressiveness: The orientation (and position) of the hand is used expressively in a wide variety of functions including gestural emphasis to speech, musical performance, drawing, painting, and drafting. *The task requires no expressive capabilities.*

Adaptability: The orientation (and position) of the hand is used for a variety of everyday purposes and so can be rated with a high level of adaptability. *The task has only one mode. Adaptability is not relevant.*

Familiarity: The orientation (and position) of the hand is used in many everyday functions and so is very familiar. *The high task analogy and familiar movement give this hand action a highly favorable recommendation for use for this task.*

Commentary

When the design method is applied to this task and the results analyzed, it indicates that whole-hand input will provide only a small improvement to dials in this task. There are three degrees of freedom involved in the task. In some cases they can be coordinated by the glove for improved performance, and in other cases the ability to decompose the degrees of freedom is advantageous, and so the dials will be better.

Resolution is important for accurately matching the rotations. The noise in the Polhemus reduces resolution to a level close to what is necessary for the task, while the dial resolution is more than sufficient. Thus, there will be times when the noise of the Polhemus hinders the performance of the task, and dial performance will be better.

Because speed of performance is the measure used for the task evaluation, the freer motion of the hand will allow for faster rotations whereas the dials are limited by trade-offs between control resolution and number of turns per degree of task angle. Therefore, for cases where the solution is clear and decoupling the degrees of freedom is unnecessary, the glove will show better performance than the dials, and vice versa in cases where the user has to resort to decoupling the degrees of freedom.

Although the orientation of the hand is one of the best whole-hand input methods for specifying orientations, it is (as is any other part of the hand) not well suited for 360° rotations. Theoretically the user never has to rotate the control stick more than 180° on any axis. However there are times when the hand gets into an awkward position and the user has to open the fist and “regrasp” the object from a more comfortable position. This takes a little time and so hinders glove performance.

The decision to use position control for this task rather than rate control was made based on the precision requirements of the task, the benefit of execution speed, and the fact that the control is generally localized to rotations of less than 180°. Rate control would be helpful for large rotations, but carries with it the difficulty of having both rapid rotations and precision alignment. Variable rate control could be used, but would not completely solve the problem and would complicate the interface. When rate control was used in other, informal tests, users often overshot the goal, oscillating back an forth to narrow the error, or had to approach the goal slowly, sometimes with fits of smaller and smaller motions to avoid overshoot. The use of position control in this task permits both fast and accurate rotations of the control stick.

In terms of naturalness of the mapping from the task to the hand, whole-hand input has a slight advantage. However, there are few degrees of freedom to coordinate, and precision limitations in both the hand and the glove device will reduce this advantage.

Putting these factor together leads to an estimate that there may be an improvement in using whole-hand input for this task, but that the improvement will be slight, mainly coming from increased mobility and speed of the hand over the dials. Experience has also shown the three-dimensional rotation tasks are highly dependent on the spatial skills of the user, and their ability to interpret depth cues in three-dimensional computer graphics.

Test results and discussion

Most subjects were able to master both tasks within the five minute practice period although some had exceptional difficulties in hard to reach orientations: mostly vertical orientations and horizontal orientations in which the stick pointed towards the user. These positions tended to bring the walker's arms into regions of singularity more than others. Subjects tended to be able to reach easy orientations faster with the glove and difficult orientations faster with the dials. One reason was that the steadyness and decoupling of the degrees of freedom of the dials helped in areas of arm singularities, while the noise of the Polhemus made glove input to the task harder. As subjects became more experienced, they became better at managing the glove, even in these areas of singularity.

Many subjects would test the effect of a dial by turning it a little ways, or rotate the hand experimentally, operating in a tight experimental feedback loop to get the correct control and orientation. The fast response of both the dials and glove made this a viable strategy for solving rotation problems, and was used more often then thoughtful analysis of the current position and predicted system response as to what motion to make, or dial to turn, to achieve the desired result.

The statistical results of the evaluation are shown in Figure 12. One can observe that, with the exception of one subject (who showed poor glove performance throughout the evaluations), every subject scored slightly better with the glove than with the dials. However, in contrast to the walking task (Section 7.1) and the path-following task (Section 7.3), the data is spread out along the line of equal performance. This indicates that personal ability with the task itself was a predominant factor in task performance, and that subjects who did well with the glove did well with the dials. That the data predominantly lies above the line of equal performance indicates that subjects do slightly better with the glove than with the dials.

A few experienced DataGlove and interactive three-dimensional computer graphics users performed this task as well. Their performance with the glove and with the dials was approximately twice that of the novice average for those devices, respectively. A t-Test using a pooled estimate of σ^2 shows that the slight difference between glove and dials is valid to at least a 95% confidence interval. This, taken together with the spread of the samples along the line of equal performance, validates the hypothesis that the glove will have a slight, but only slight, advantage over the dials, and that other factors predominate in the performance of the task.

figs/rot-results.eps

Figure 12: *Results of orientation task* Shows the data from the orientation task in units of average orientation matches per second. The dashed line shows the line of equal performance, where the subject would have performed equally well with either device. Points above the line represent subjects that do better with the glove than with the dials.

7.3 Path-following task

The goal of this evaluation was to test a task for which the design method does not give a recommendation that the task should use whole-hand input and predicts superiority of conventional devices over whole-hand input. The task chosen was one in which the user guides the walker in following a prescribed path. Nine subjects participated in this evaluation.

The task and experimental method

figs/path-task.eps

Figure 13: *The path-following task* The six-legged walker and a path to follow are viewed from above. On the right are illustrated the whole-hand and conventional device control inputs. The subject forms a fist or presses a button to start the walker moving at a constant speed. Palm or dial orientation indicates the compass direction for the walker.

The six-legged walker is capable of walking on its own using a simple oscillatory gait generator that coordinates the legs to follow a steady tripod gait. For the purposes of this experiment, the walker's variable speed control was disabled; the walker either moved forward at a constant rate or did not walk at all. Steering of the walker was independent of forward motion so that the walker could be oriented even when stopped. Orientation was controlled by specifying the compass direction. This is in contrast to a rate control which controls turning left or right. The walker had no maximum turn rate and would immediately orient to the direction indicated.

The whole-hand input control for the task used the fist posture as a button to enable walking. As long as the hand was held in a fist, the walker moved forward. Opening the hand stopped the gait generator. The direction of the walker was indicated by the hand's palm direction in the horizontal (x-z, or *left-right/forward-back*) plane (see Section 9.6). Thus, the walker turned to face in whatever direction the user's palm (open or closed) was facing. The conventional input controls used a button to indicate walking—the gait generator was enabled as long as the button was held down—and a dial to indicate the direction of travel. The dial had a one-to-one correspondence with the walker direction so acted as an absolute direction control.

Subjects were asked to guide the walker along each of four reference paths (Figure 14). The paths were presented in a random sequence, first for a measured practice run, and then a second time for a measured "real run." The best of the two runs was chosen for the task evaluation. Measurements were taken every three frames (with the system running at ten hertz) and consisted of the x and y position of the walker.

The success of the run was measured by taking the area between the reference path and the path followed by the user. The smaller the area of the region of error, the more successful the run. This method eliminated timing effects, since some subjects stopped and started more often than others. Figure 15 shows a typical run and the area calculated as the region of error.

Application of the design method

Appropriateness of whole-hand input:
Is the use of whole-hand input appropriate and beneficial to the application task?

Naturalness

Are pre-acquired sensorimotor skills useful for controlling the task?

> Probably. The task does not have a very complex control structure, yet requires skilled fine control to perform well. Existing sensorimotor skills for fine hand control may be useful for the task.

Are existing hand signs skills useful for controlling the task?

figs/path1.ps

figs/path2.ps

figs/path3.ps

figs/path4.ps

Figure 14: *Reference paths* The four reference paths subjects followed for the path-following evaluation. The walker is shown at the beginning of each path. All paths were followed starting towards the right.

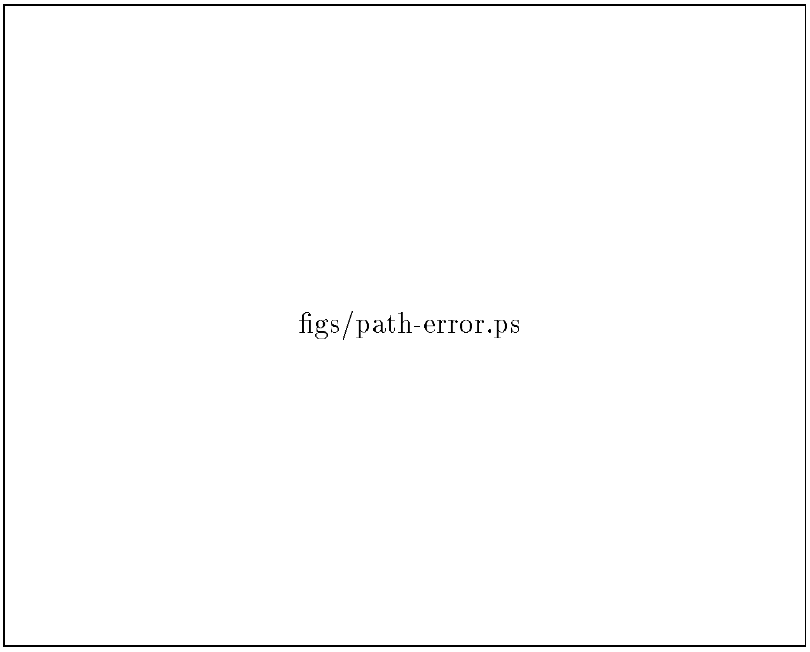

Figure 15: *Region of error* This shows in gray, the region of error calculated for one of the runs in the path-following task.

> No. The way the task has been set up, there is no need for more than the one hand sign. If the orientation task were to use a different style of control, then perhaps existing hand signs such as those used in traffic control and commonplace directional communications would be useful.

Is the absence of an intermediary device useful for controlling the task?

> Unknown. The use of an intermediary device may actually provide useful constraints to the task (such as tactile feedback from device friction, constraining motion to the one important degree of freedom, and providing a resistive force to steady the hand).

Can task control map well to hand actions?

> Not well. The task requires continuous 360° rotational control. There are few hand motions that have a continuous 360° range without singularities (aside from shoulder rotations in the vertical plane). It is possible to use the hand in

a manner that mimics using a device, such as a crank or a wheel, to generate continuous rotation. This would be useful if other parts of the task were likely to use whole-hand input. However, for this task alone, there might as well be a real crank, wheel, or dial with its beneficial constraints.

Adaptability

Are diverse modes of control used in the task?

Not for this specific subtask. Only one mode of control exists for this task. However, because this is a sub-task of a larger application with many modes of control, the answer for the whole application is yes.

Is it important to be able to switch between modes of control rapidly, and smoothly (i.e., with a minimum of distraction from the task)?

Not for this specific subtask. See above. For the whole application however, it is important to switch smoothly between the modes for efficient execution of the task.

Coordination

Does the task require the coordination of many degrees of freedom?

No. There are two degrees of freedom of the task. One is binary—walking on/off—and the other is direction control. They require little or no coordination.

Real-time control

How important to the task is continuous monitoring?

Very important. The walker is always moving. To follow the path, the user constantly must be making adjustments to the heading.

How important to the task is rapid user response?

Important. The user must remain alert and able to respond to curves in the path. However, none of these are unanticipated. There are no environmental effects to which the user needs to react.

Taxonomy – input style:

What style of whole-hand input is intended and/or seems to be best suited for this task?

	Direct	**Mapped**	**Symbolic**
Continuous	Continuous/Direct	Continuous/Mapped	Continuous/Symbolic
Discrete	Discrete/Direct	Discrete/Mapped	Discrete/Symbolic

The specifications of the task require a style of interaction similar to the orientation task. Conceptually it is a Continuous/Direct task, while computationally it is a Continuous/Mapped task, since the walker does not actually rotate on a point, but executes the correct stepping patterns to turn the body based on the subject's supervisory control of direction.

Evaluation guide

Task characteristics and requirements

The task (as presented) has two functional units: orientation of the walker and enabling or disabling of walking. The most important function is the orientation of the walker. This is the one analyzed in the following evaluations. The other function has a parallel analysis which has been omitted for the purposes of this text.

Degrees of freedom: The task has two degrees of freedom: the binary switch turning walking on or off, plus the continuous direction indicator.

Task constraints: There are no particular constraints on the task.

Coordination: The two degrees of freedom are independent and require little or no coordination.

Resolution: The walking switch requires 1 bit of resolution. The directional control resolution is dependent on the accuracy with which the task must be accomplished. It must have at least 2 bits of accuracy to specify up, down, left, or right. A desirable minimum accuracy would be on the order of 8 bits to specify direction in 2° increments.

Speed: The walker must be able to turn fast enough to follow the path. Given its forward speed, this is approximately 180° per second.

Repeatability: Repeatability for this task specifies how closely the path is to be followed, i.e., how accurately the user must be able to achieve a particular direction. There is no specification of this in the task description for this experiment. In fact, it is one of the requirements that this experiment could (but does not) measure.

Steadiness: Steadiness is important to maintain particular headings along straight sections of the path and to maintain steady control over the walker. The steadiness requirements for the task are on the order of seconds.

Endurance: Navigation of a path takes between 60 and 90 seconds. This is done eight times over the course of approximately thirty minutes. The user can rest between paths.

Expressiveness: There are no requirements of expressiveness for this task.

Modality: This task has only one mode. However, if part of a larger application, there must be a way to shift out of the one mode.

Task analogy: Most guiding tasks, such as steering a vehicle, are performed by rate control, not compass direction. However, people are not unaccustomed to indicating absolute direction and can use these skills in the task.

Hand action capabilities

As was done with the previous two experiments, only the evaluations for the chosen hand action, *palm orientation (with closed fist)*, are presented here. Again, informal observations are used for some of the measures in the absence of clinical data. Following each entry in the hand action capabilities is a short comparison to the corresponding task requirement.

Degrees of freedom: This hand action has essentially one degree of freedom: the orientation of the palm. *This matches the directional degree of freedom of the task.*

Hand constraints: The hand cannot be rotated through 360° in any one axis without rotations in the other axes to overcome the kinematic constraints of the wrist, elbow, and shoulder. Palm orientation is mainly accomplished through wrist rotation with a range of 140°-160°, (American Academy of Orthopaedic Surgeons, 1988) Range limits in palm direction and joint singularities can be overcome with mild, but awkward contortions combining wrist flexion/extension and raising or lowering of the elbow. However, the awkwardness of these actions slows down the speed of rotation. In addition, palm *forward-back/left-right* direction has a singularity at high and low elevations (see page 157) where the direction is undefined.[7] Note: Other hand postures were considered. *Pointing* in the correct direction seemed the most intuitive method for indicating direction, but having a single finger extended while the others are flexed, strains the ligaments as the wrist is brought to its joint limits. This effect is not as pronounced as when the fingers are all opened or all closed, so the fist was chosen for the task. *The rotational constraints of the hand make the task difficult to control with this method of whole-hand input. However, based on evaluations of other whole-hand input methods and because continuous 360° control is required, this is the best method for the task short of mimicking the action of manipulating a wheel or dial. The limitations can be overcome, but at the expense of speed of task execution and user response—important factors in this task.*

Coordination: Although only one degree of freedom is being actively controlled, the user must coordinate several hand/arm degrees of freedom to achieve specific orientations. *The task has only one degree of freedom, so no coordination is required. That the hand needs to coordinate degrees of freedom to accomplish the task may be a disadvantage in using whole-hand input here.*

Resolution: Unknown. The wrist, elbow, and shoulder determine the orientation of the hand (relative to the body) and thus determine the resolution with which the hand can be oriented. *The resolution requirements for the task are not stringent and the hand should have enough resolution for adequate performance.*

Speed: Unknown. Informal observations put the speed of a rapid wrist motion at 300–400 degrees/second. *This should more than satisfy the task requirement for approximately 180° per second.*

Repeatability: Unknown. *Informal observations and previous experiments by Drucker (1990) put the ability of the hand to perform this kind of task (and thus repeatability of hand orientation) within the requirements of the experiment.*

[7] The solution of filtering out the singularity was not thought of until several subjects had been tested, so the method of only warning subjects of the singularity was continued for the entire experiment.

Steadiness: Informal observations indicate that the hand can be held unsupported in place for 30 to 60 seconds before fatigue causes unsteadiness. If the forearm is supported the time period is greater. *This is well within the steadiness requirements for the task.*

Endurance: Informal observations indicate that orientation of the hand can be performed on the order of hours given occasional periods of rest. This is probably due to the extensive use of the hand for everyday tasks. *This is well within the thirty minutes required for the experiment.*

Expressiveness: The orientation of the hand is used expressively in a wide variety of functions including gestural emphasis to speech, musical performance, drawing, painting, and drafting. However, reducing this to one value, the direction of the palm, severly limits expressive capability. *The task requires no expressive capabilities.*

Adaptability: Orientation of the hand is used for a variety of everyday purposes and so can be rated with a high level of adaptability. *The task has only one mode. Adaptability is not relevant.*

Familiarity: The orientation of the hand is used in many everyday functions and so is very familiar. *Although used for many tasks, hand orientation is not often used as a directional controller in the manner prescribed for this task. It may be possible for users to adapt the hand directional skills of everyday use to the particulars of this task.*

Commentary

The design method indicates that whole-hand input is not well suited for this task, although it can be used at the expense of task performance as compared to conventional input methods. The task nominally has two degrees of freedom that require little coordination but some precision. For a task of so few degrees of freedom not requiring coordination, whole-hand input is not expected to provide significant improvements over conventional methods. Likewise with naturalness. Experience indicates that a dial is a natural steering device and that free-hand motion has little advantage in this area.

If whole-hand input could provide an improvement in naturalness then its use would be advantageous, but experience indicates that the task of turning is performed with the dials naturally, and whole-hand input does not have an advantage in this area.

The direction control requires 360° rotation, a bane to whole-hand input. Although whole-hand input can handle 360° rotations if controlled by rate, the task required a directionally-absolute scheme of control. There are strategies that allow users to continuously rotate the palm so that it points correctly. They involve slight contortions of elbow, but these are not intuitive and many people have difficulty in doing them reliably without a lot of practice. (This hypothesis was borne out by the novice users who, in the twenty or thirty minutes that they were performing the task, found that these motions most likely got them in trouble.) The precision of the task is within the precision of the hand, but the hand is not as steady as the dials, nor is it easy to hold in place. Path-following requires a steady control. For these reasons the dials should show an advantage.

Task results and discussion

Subjects reported that the task was fairly simple to perform with the dials. However, they had varied experiences with the glove as evidenced by the greater variation in glove performance (Figure 16). Most found the task more difficult with the glove control for the reasons hypothesized above. Error recovery was more difficult with the glove, because rapid course corrections, fostered by sensor lag, tended to overshoot the path, leading to unstable oscillations as the subject tried to get back on course. Furthermore, subjects got into trouble at extreme rotations of the wrist, where they had little mobility with which to correct the situation. With the dial, subjects seemed to be steadier and more relaxed, and achieved better performances.

The statistical results of this task are shown in Figure 16. It is evident that all the samples fall at or below the line of equal performance indicating greater error in path following with the glove than with the dials.

figs/path-results.eps

Figure 16: *Results of path-following task* Shows the data from the path-following task as one minus the normalized area of error between the reference path and the user's path. Note that the larger the number, the better the performance. The dashed line shows the line of equal performance, where the subject would have performed equally well with either device. Points below the line represent subjects that do better with the dials than with the glove.

8 Testbed for Whole-Hand Input

The testbed for whole-hand input is a computer framework that allows the linking of whole-hand input devices to interactive graphical simulations. The purpose of the testbed is to experiment with whole-hand input control methods using a variety of whole-hand input devices in different task domains. At one end of the testbed are a variety of whole-hand input devices, at the other end are a set of tasks simulated with interactive computer graphics. In the middle, at the core of the testbed, are a series of software routines which transform hand and finger actions into task control signals. These routines embody the different control methods to be explored. Consequently they must be easy to create and modify.

The testbed is implemented on a Hewlett Packard 9000 series 835 workstation running HPUX, a version of the UNIX[1] operating system. Graphics are handled by an HP Turbo-SRX graphics accelerator card installed in the workstation.

The testbed is conceptually divided into four sections as shown in Figure 17.

figs/testbed.eps

Figure 17: *Testbed* Information flow for testbed for whole-hand input.

[1]UNIX is a trademark of AT&T Bell Laboratories.

1. Whole-hand device capturing hand actions. The whole-hand input devices currently that can be connected to the testbed include the VPL DataGlove, the Exos Dexterous HandMaster (DHM), and the Mattel Power Glove. Conventional input devices also are available and include a set of six dials, a 32-button box, a mouse, a data tablet, and a Spatial Systems Spaceball.

The three whole-hand input devices are connected via serial communication lines to an HP Real-Time Interface card (RTI) sitting on the backplane of the workstation. The RTI has it's own processor which performs serial I/O asynchronously from the main workstation CPU, continually reading and storing the device output records. The workstation CPU retrieves these records from the RTI as it needs them.

The DataGlove interface unit is connected directly to the RTI via an RS-422 line. The Power Glove is connected to a special-purpose interface that converts the Power Glove's proprietary and encoded Mattel/Nintendo output to computer readable ASCII RS-232 signals.[2] These are sent to the RTI. The DHM is connected to an analog-to-digital conversion card on a 80386-based PC. Initial processing is performed on the PC. Data are then sent via RS-422 serial communication to the RTI.

2. Abstraction routines analyze the device sensor data and convert them into generic hand actions (degrees of freedom, continuous features, and discrete features—page 80). There are at least two levels of routines, the first of which is specific to the device and transforms the raw sensor values into a normalized whole-hand input space representing the degrees of freedom of the human hand. The second and subsequent levels of abstraction are device independent and derive continuous and discrete features from the degrees of freedom. The software in these routines includes code which takes first, second, and third order derivatives of the degrees of freedom of the hand, performs feature analysis (Rubine, 1991) on hand actions, and provides posture and gesture recognition. The three classes of features are passed in device-independent data structures to the interpretation routines. Section 9 discusses the implementation of whole-hand input abstraction routines.

3. Interpretation routines transform hand actions into task control signals. They can be any variety of code such as programmed algorithms, finite state machines, and

[2] The serial interface was made available to this work by Abrams/Gentile Entertainment, Inc., the designers of the production version of the Power Glove. Less then two hundred of these interfaces have been manufactured and distributed as of the writing of this dissertation.

mathematical relations. Some examples might be as simple as linking the flex of the index MP joint to the rotation of a joint on a robot arm; or the more complex control of a small gripper by the opposition of the thumb and forefinger; or the highly complex simultaneous control of the speed, attitude, heading and systems control of an unmanned submersible.

4. Task simulations are performed with an existing computer program called *bolio.* Bolio was developed by the MIT Media Lab's Computer Graphics and Animation Group for rapid prototyping of interactive simulations using real-time computer graphics (Zeltzer, Pieper, and Sturman, 1989). It maintains and displays a database of graphical objects that can be continuously manipulated by externally running routines. The testbed acts as an external routine to bolio's database, effecting changes to the objects in the simulation as determined by the task control signals. For example, a robot manipulator might be built from a set of related graphical objects. Control signals would change the angles between the graphic objects forming the links of the manipulator arm. Section 10 discusses three prototype applications built using bolio.

9 Implementation of whole-hand input abstractions

This section covers techniques related to developing a library of whole-hand input interpretation routines that take in raw whole-hand input sensor data and output abstracted whole-hand input parameters, postures, and gestures, essentially developing the foundations for an abstract whole-hand input device type. One of the contributions of this dissertation is the comprehensive presentation of these techniques, old and new, so that researchers and developers can implement a whole-hand input library for their own use.

There is a variety of sensor data that can be captured from the hand. For the fingers, sensors can record the flex of the fingers as a whole (the Power Glove), the flex of individual joints (the DataGlove, DHM, and the CyberGlove), the absolute position of the fingers (Selspot), silhouette edges of the hand (Krueger and Mandala systems), or discrete events such as finger touches (Grimes). For hand position, sensors can record the absolute position of the hand in a variety of ways (Selspot, Polhemus, Bird[1]), or relative from the body using bodysuits (VPL) or braces with angle sensors.

For the purposes of the following discussion it is assumed that the sensor data is derived from finger-flex sensors and a three-space (six degrees of freedom) position/orientation sensor tracking the hand. The systems commercially available at this time conform to this convention. Although the techniques described below specifically address these types of systems, they can be adapted to other forms of data retrieval.

9.1 Conditioning flex sensors

Most sensor devices have inherent noise problems. Some "condition" their sensor signals before sending them out to the host processor, some do not. If the signal coming to the host processor shows noise, then digital filtering must be performed on the signal. For instance the DHM uses 12 bit analog-to-digital converters. Of the 12 bits the host receives for each sensor, the bottom two or three bits are mostly noise from the electronics and micro-tremors in the finger joints. The testbed implemented for this work removes this noise with a right-shift of three bits, yielding 9 bits of precision.

[1]The Bird™ is a three-space tracker made by Ascension Technologies, similar to the Polhemus.

9.2 Calibration and normalization of flex sensors

Different whole-hand input devices have different ranges and precisions of sensors, and although hands have different ranges of motion it is advantageous to adopt a set of standard conventions that can be applied to all hands. Calibration and normalization procedures transform device sensor readings into a standard form that is similar for all hands and can be dealt with by all subsequent levels of whole-hand input software.

The convention used in this work characterizes each joint flexion as a single-valued continuous quantity. Thus, hand shape can be characterized as a vector in a 23 dimensional space, each dimension representing a different flex. Flexions are normalized to a value between 0.0 and 1.0, where 1.0 corresponds to full flexion, and 0.0 corresponds to full extension.[2] Although many people can hyperextend their MCP joints (bend the joint past zero towards the back of the hand), functionally, they consider the finger as simply "outstretched" when it is in this position. Except in conditions where the accurate angle of the finger is important, hyperextension can be treated as the finger straightened normally. In conditions where the actual angle is important, hyperextension can be treated as negative values of flexion. Ab/adduction of the thumb and MCP joints is also normalized from 0.0 to 1.0 where 0.5 is considered the center or relaxed position.

The advantage of this normalization is that postures and gestures specified in the normalized space work for any person that has been properly calibrated for a particular session with a particular device. For instance the "pointing" posture is characterized by values of 0.0 on the index MCP, PIP, and DIP joints, and 1.0 on all other joints (ignoring ab/adduction angles).

[2] The technique of normalizing flexion was independently developed for use with the DataGlove at the NASA Ames Research Center (Fisher et al., 1986) but never reported in the literature (personal communication, Steve Pieper).

figs/flexion.eps

Figure 18: *Normalized flexion* Shows the normalized values assigned to flexion and ab/adduction of the finger joints.

Key posture calibration

There are several methods available for device calibration. Most rely on interpolating two sensor readings from distinct finger positions. The basic method, that suggested by VPL in the DataGlove User's Manual and used in their Apple Macintosh-based user software, takes readings of key-posture finger positions and fits them with a continuous function. In theory the DataGlove sensors have an exponential response to the respective joint angle.

$$r = k_1 e^{-k_2 \alpha} \tag{1}$$

r is the joint angle, α is the device sensor output, and k_1 and k_2 are constants.

The constants k_1 and k_2 can be found with two sampled data points per sensor. The first sample reading VPL takes is with the fingers fully extended. These readings are used as the minimum flexion angles and assumed to be 0.0°. The second reading is with the thumb relaxed and the fingers fully flexed in both the MCP and PIP joints. The angles are assumed to be 90° for the MCP and PIP joints. A final reading is taken with the thumb bent inward to the palm and the fingers relaxed and out of the way, to get the thumb angles—45° at the MP joint and 90° at the IP joint. The exponential constants are computed to get an interpolation function that converts sensor readings to joint angles. Wise et al. (1990) found the DataGlove sensors and interpolation method was accurate to within approximately five degrees of flexion, or approximately 4-5% of the total range.

Research for this dissertation has found that for many whole-hand input techniques an accurate angle is not strictly necessary. Repeatability and dynamic range (available bits per sensor) are much more important. Linear interpolation is more than adequate for most applications, especially for the DataGlove, given the relatively low precision of its sensors. Other devices, such as the DHM and Power Glove also work adequately with linear interpolation functions.

In practice however, a linear function is faster to compute and has been found to be adequate given the relatively low precision of the DataGlove sensors. The minimum and maximum sensor values read during calibration are normalized to 0.0 – 1.0 by the linear function (2).

$$value = (sensor - min)/(max - min) \tag{2}$$

A copy of this function with different min and max values is associated with each sensor. In theory any two key postures of known angles can be used. This is the essence of the template approach (below). If absolute angles are not relevant (as is the case with much of the whole-hand input library presented here) then the 0.0 to 1.0 range is sufficient.

Continual calibration

An alternative to key postures is to calibrate continuously; monitoring sensor input and keeping track of the minimum and maximum readings. If a new minimum or maximum occurs then the calibration equation (2) is given the new parameters. The advantage of this scheme is that users need do no more than flex their hands a few times after initializing the minimum and maximum settings. Experience has shown that within a few seconds of use, calibration settings are attained that last the remainder of the session.

One drawback of this method is sensitivity to sensor noise (either mechanical or electrical) that can cause spikes in the minimum and maximum values. A low-pass filter applied to the values before they are compared to existing minimum and maximum values can help to eliminate this problem. Another solution (the one used in this work) is to provide a short period (ten to fifteen seconds) of calibration during which the user flexes their hand through its total range of movement. This has the dual advantage of making the calibration step easy for the user, and fixing the calibration for the remainder of the session. At any time, if the calibration is unsatisfactory, the user can initialize the maximum and minimum values and do a quick recalibration.

Template-based calibration

If precise angle readings are important then it is possible to use lengthy calibration methods requiring angle templates to fix sampled positions. This was the approach taken by Hong and Tan (1989) and Burdea et al. (1992). Hong and Tan (and from their work, Burdea) used mechanical means to set the finger joints to known angles (0° to 90° in 5° increments) to accurately calibrate VPL DataGloves. Based on experimental data relating finger angle to sensor reading, they determined that the following formula (3) properly represents the relation between DataGlove sensors and joint angles, where r is the sensor reading, α is the computed joint angle, and a, b, and c are coefficients found by a least-squares fit of experimental data.

$$\alpha = ar + b + c \log r \tag{3}$$

They report that this calibration takes about fifteen minutes and is dependent on the wearer and the particular DataGlove being used. They found significant variations in the sensor properties among the three DataGloves they had at their disposal. Unfortunately, they found that the sensors on the DataGlove were highly correlated with each other and that a further step was necessary for accurate sensor-to-angle calibration (described below).

Adjusting the calibration parameters

It is often the case that the calibration is satisfactory for some joints but not for others. In many of these cases the calibration is close, but a small adjustment would be preferable. The capability should exist to make fine adjustments on individual sensor calibration parameters, either on the minimum or maximum values directly, or on offsets applied to these values.

Finger joint and device sensor crosstalk

Finger joints are not strictly independent. They function with and are supported by a complex, interconnected web of muscles, tendons, and other connective tissues. Neurologically the fingers are also dependent with joints sharing common neurologic and muscular activators. This fact is clearly evident in the workings of every hand. For instance, most people cannot bend DIP joints independently of PIP joints, and some people cannot bend their pinkie PIP joint independently of their ring PIP joint. The interrelations of these structures is not completely understood, and research in the area (for instance Guidera (1981)) is still developing new understanding of the hand's structures.

The interrelation of the hand's structures is important in the design of whole-hand input methodologies as discussed in Section 6. The focus of this section is to interpret the sensor information from the hand. The constraints on hand motion are relevant when designing and developing posture and gesture recognition techniques (a posture requiring one finger open while the others are closed needs to take into account that the open finger cannot be as widely opened when the others are closed as when the other fingers are opened—thus the posture may only require a normalized MCP value of .3 or less, rather than .1 or 0.0), but joint flexions have distinct values, regardless of the constraints imposed upon them, and the sensors and calibration should reflect this.

If the sensors themselves exhibit crosstalk, i.e., each sensor reading is dependent on the state of the other sensors (as is the case with the VPL DataGlove), then the nature of the crosstalk must be accounted for in the calibration to get an accurate representation of the joint flexions from the sensor readings. The nature of the VPL DataGlove sensors, for instance, make this a particularly difficult problem. The fiber-optic sensors are sensitive (yet non-discriminatory) to bend in any direction and the flexion of any one MCP joint pulls laterally on neighboring MCP flex and abduction sensors, improperly registering flex in the neighboring joint.

Hong and Tan (1989) investigated a calibration technique to account for this behavior in the VPL DataGlove. After an initial calibration step (described above), they fixed one sensor to a known angle. Then they systematically varied the other sensors keeping a record of the known angle and observed sensor readings. Then they fixed the first sensor to another angle and again varied the other sensors. By repeating this process for a series of angles on a specific sensor, a function can be derived that eliminates the correlation factors from the other sensors. Theoretically, that one sensor then has a calibration function that yields an accurate angle from any set of sensor readings across the DataGlove. Using this sensor as a basis, the process is repeated for the other sensors.

Needless to say, Hong and Tan report this process as extremely tedious, time-intensive, and error-prone. However they feel that the process can be done once with an average hand and stored in a permanent table. A small set of control parameters then can be applied to the calibration functions to modify the results to fit individual hands, individual DataGlove sensors, and the slight drift of the sensors over the duration of a session. They write that few subjects have been tested, and that more experience is needed before a firm conclusion can be reached as to the feasibility of this method.

Hong and Tan's conclusions should apply to any device with sensor crosstalk. Unless the function for describing the crosstalk between sensors of a device is well known and user independent, a calibration process that accounts for it on an individual basis is likely to be tedious and error-prone. This is purely a function of the complexities and combinametrics of finger-joint motion (there are 256 combinations of opened or closed MCP/PIP joints alone). If cross-talk is a serious problem that must be solved, then perhaps it may be appropriate to build electro-mechanical devices that quickly cycle the fingers through known flexion angles, recording simultaneously the finger angle and whole-hand input device sensor reading, actively generating cross-correlation tables for the various combinations of joint flexion.

9.3 Flex transformation functions and tables

Flex transformation functions and tables transform the normalized joint flex space to parameters in the control space for a particular application. As an example, an exponential function is used to improve parameter sensitivity for musical performance in *Bug-Mudra* described in Section 10.4. Mapping functions can take any form. They can be a simple as a linear transformation $value = A * flex + B$, such as is used in the low-level walking controls described in Section 10, or more complex, involving a variety of functions across different ranges of motion, or filters smoothing or enhancing the data. In a preliminary test of subjects' abilities to use the six-legged walker's manipulators at the low-level of control (described on page 172), it was found that applying an averaging (IIR) filter to the flex values improved performance as it gave the manipulators a slightly damped characteristic.

If the transformation function is complex or computationally expensive then it may be advantageous to build transformation tables. The cost of these tables is the memory required. However, memory is often cheap and plentiful, and processing power can be better used in other areas of the application. As an example, the exponential table used in *Bug-Mudra* used a reflected exponential to transform values from the DHM to the parameter space. Since the DHM outputs 12 bits of data (although all are not useful) for each sensor, tables of 4096 entries per sensor were sufficient to map the entire hand space. The reflected exponential function was evaluated for each of the 4096 possible input values and the results stored as 4-byte values in the transformation table. Since only four sensors were used, the memory requirement was 64K. Even if all the sensors were used, only 327K of memory would have been required.

9.4 Posture recognition

As defined on page 80 posture refers to hand shape and position, as distinct from gesture and hand motion. There are many methods for recognizing specific hand shapes, each having advantages and drawbacks.

Hardware solutions: Digital Data Entry Glove

Probably the earliest posture recognition was by Grimes (1983) whose Digital Data Entry Glove was designed specifically for recognizing the signed alphabet. Carefully placed sensors registered fingertip contact, flexion of specific joints, and hand attitude. Posture recognition was hard-coded into the electronics of the glove; a particular combination of sensor readings produced an ‘A’, another a ‘B’, and so on. The advantage of this technique is rapid and robust posture recognition. The disadvantage is inflexibility, in that only those postures it was designed for can be recognized.

Table lookup

VPL’s posture recognition software was probably the next entry in the field (VPL uses the term *gesture recognition*). They use a lookup table method. For each posture, each sensor has a range of values that are valid for that posture. For instance, a fist would be characterized by the range of values the sensors return when the hand is closed. At each sample time the sensor readings are compared with the values in the posture tables. If each sensor reading falls within the range of the corresponding table entry then a posture is recognized. VPL's software also provides hysteresis values for each sensor entry to widen the range of the match once the posture is recognized. This helps the user hold a posture once it has been recognized. VPL provides a *gesture editor* that permits users to generate and/or tune posture tables by hand (Figure 19).

The advantage of the VPL method is that it is simple and flexible enough to generate and recognize postures easily and quickly. The representation of postures is easily understood and can be modified by the user. The drawback of this scheme is that in practice, the range of each table entry must be quite wide, up to 30% of the total flexion range (plus hysteresis), due partly to inaccuracies in the VPL DataGlove and partly due to imprecise user postures. With more than about ten postures, there occurs a wide range of overlap among posture tables causing many hand shapes to match more than one table entry. With a more accurate hand device this will be less of a problem

figs/vpl-gesture-ed.eps

Figure 19: *VPL gesture editor* The gesture range is in dark grey, and the hysteresis range is in light grey.

VPL's "gesture" editor generates tables that can be downloaded to the glove interface device which performs the posture recognition in firmware. Given a gesture table, the device will return a gesture byte along with the sensor readings in each update frame.

VPL's table lookup scheme is simple to implement in software and, despite its drawbacks, is useful for the ten or fifteen common postures formed by combinations of fingers opened or closed (e.g. fist, pointing, "peace" or "victory," "thumb's up," "pistol," "the finger," and so forth). As these postures are sufficient for many applications, it is one of the posture recognition schemes used in the whole-hand input testbed developed for this dissertation (see "Simple method" below). The actual algorithm used is as follows:

To record a posture, the user forms the posture and approximately fifty samples of the sensors are recorded (about three seconds worth). For each sensor, the minimum and maximum sampled values are placed in a table. To these are added (or subtracted) an additional factor to account for device inaccuracies. The posture recognition code compares incoming sensor values in `flex[i]` with values stored in the posture tables, `min[p]` and `max[p]`.

```
posture_recognized = -1;
 /* loop through all postures, p */
for ( p=0; p < NUM_POSTURES; p++ ) {
    hysteresis = (previous_posture==p) ? hyster_value : 0.0;
     /* loop through all sensors, i */
    for ( i=0; i<NUM_SENSORS; i++ ) {
       if ( flex[i] < min[p][i]-hysteresis ||
            flex[i] > max[p][i]+hysteresis )
          break;  /* out of range, no match */
    }
    if ( i==NUM_SENSORS ) {
         /* all sensors fall with in the range of the table entries */
        posture_recognized = p;
        break;
    }
}
previous_posture = posture_recognized;
```

A variation on this method is to extract an average and standard deviation from the sampled sensor values for each posture.[3] Posture recognition is accomplished when each

[3]Suggested in personal communication with Tom Zimmerman, one of the developers of the VPL DataGlove.

of the sensor values fall within the appropriate standard deviation of the corresponding average value in the table. The sensitivity of the posture recognition can be adjusted by varying the standard deviation (or error factor). The same error factors can be given to all the sensors, a separate one maintained for each, or a sum error computed.

$$error = \sum_{i=sensor_0}^{sensor_n} flex_i - average_i \tag{4}$$

The sum error method recognizes a posture when the sum of all the errors falls below a certain threshold. This recognition method is useful to permit some variability or slack in the overall precision of the posture.

Simple method

A third, and the most simple modification of the table-lookup method stems from the observation that ninety percent of the useful postures (especially with the table-lookup method) involve some combination of the finger joints either fully extended or fully flexed (Figure 20). If flex values are normalized between 0.0 and 1.0, a set of common, *joint-limit* postures can be hard-coded simply by looking to see if a joint is less than 0.2 or greater than 0.8 (these are the values used in this work). For instance, a fist is characterized by all values greater than 0.8. A pointing gesture is a fist but with index flexions less than 0.2. Certain sensors can be ignored in this scheme. For instance with the pointing posture, it may be appropriate to ignore the thumb; some people tuck it in, some people don't. In addition, it may be sufficient to look at only one of the three knuckles of a finger. "Pointing" recognition code can be something like this:

```
/* flex[d][j]: normalized flex value for digit d, joint j */
if ( flex[index ][MCP] < 0.2 &&
     flex[middle][MCP] > 0.8 &&
     flex[ring  ][MCP] > 0.8 &&
     flex[pinkie][MCP] > 0.8 ) {
  posture_recognized = POINTING;
}
```

This method has the advantages of being simple to code, robust, and fast to process. Unlike the other table lookup methods, it does not require each user to train the system figs/limit-postures.eps

Figure 20: *Postures using joint limits* These are a few of the many finger postures using joint limits. Not all joint-limit configurations are achievable or comfortable, however there are enough practical postures to satisfy most applications.

for each posture. Once a device has been correctly calibrated to the user, the postures will be recognized automatically. The drawback to this method is that each posture is hard-coded into the system. However, in the development of the prototype applications for this dissertation, this has been found to be only a slight inconvenience compared to the greater benefit of not requiring each new user to generate posture recognition tables.

A compromise between the table look-up and the simple method of hard-coding the set of joint-limit postures is to use a table-lookup system that only stores joint-limit postures ($0.2 \leq flex_i \geq 0.8$), along with "don't care" indicators for irrelevant joints. This would require one training per posture for all users, not per user per posture.

Complex methods

If more than ten or fifteen postures are required, then more complex methods of posture recognition must be employed. A discussion of other methods appears in Section 5 starting on page 68. These methods are generally more difficult to code, require extended training times, and are less flexible than the methods described above. However, they tend to be able to recognize more postures and more complex postures than the above methods. The literature has no comparisons between methods, and controlled studies of robustness for each of the methods are not available.

9.5 Three-space sensor reference frames

Three space sensors often have their own "natural" reference frames that do not correspond to a convenient reference frame when the sensor is attached to the hand. It is useful to convert the default reference frame of the device to a "standard" that can be used by device-independent software. The standard chosen for this work is positive x to the right, positive y up, and positive z towards the user, when the user is facing the graphics screen. This is a right-handed coordinate system. Rotations are *azimuth*, positive around the z axis from x to y; *elevation*, positive around the rotated y axis from z to x; and *roll*, positive around the rotated x axis from y to z. The whole-hand input library assumes this reference frame. (Figure 21 shows the reference frame with the setup for the Polhemus three-space tracker.)

figs/polhemus-ref-frame.eps

Figure 21: *Whole-hand input standard reference frame* The standard reference frame used for whole-hand input is shown with the Polhemus sensor. Note that the Polhemus is mounted right-side-up on top of a table. Also note the awkward orientation of the sensor to achieve the zero-rotation position. This is because for the two three-space trackers used in the testbed, the Polhemus and the Bird, the source reference frames can be changed but the sensor reference frames cannot.

This reference frame was adopted as the zero-orientation standard for the whole-hand input software library because it coincides with the conventional standard used in matrix algebras. There is no similar convention for three-space trackers, and each brand must be transformed in a different manner to match the whole-hand input standard. For the two used in the testbed, the Polhemus and the Bird, this transformation results in an awkward zero-orientation hand position—palm towards the user, fingers right. However, this is important only to the software interpreting hand action, not to the user. Therefore, it is computationally convenient to use as a standard.

For each three-space sensing device, a different transformation must be used to rotate the default device reference frame into the desired whole-hand input reference frame. The default Polhemus reference frame is x to the user, y right, and z down as shown in Figure 22. This can be changed by a combination of physically reorienting the source and using the built-in Polhemus command to realign the reference frame. However, this command realigns only the source reference frame, not the sensor reference frame, thus the awkward zero-rotation hand position. The realigned Polhemus reference frame is shown above in Figure 21.

figs/polh-default-ref-frame.eps

Figure 22: *Polhemus default reference frame* The default reference frame for the Polhemus assumes the Polhemus source is mounted on the underside of a table. Note that when the wires coming from the source and sensor are parallel and the sensor is right-side-up, the axis align. This is the default zero-rotation position.

The default reference frame for The Bird is shown in Figure 23. To transform it to the whole-hand input standard a $-90°$ rotation about the x axis followed by a $+90°$ rotation about the y axis accomplishes the result. The Bird does not have a Polhemus-equivalent command to realign the reference frame. (Although, it does have commands to rotate the reference frame, their effect is different than the Polhemus realignment command.)

figs/bird-ref-frame.eps

Figure 23: *Bird default reference frame* The default reference frame for the Bird assumes the source is mounted on top of a table. Note, that like the Polhemus, when the wires coming from the source and sensor are parallel and the sensor is right-side-up, the axis align. This is the default zero-rotation position.

Left- vs. right-handed coordinate systems Most three-space sensor devices (including the Polhemus and Bird) use right-handed coordinate systems, while many computer graphics packages use left-handed coordinate systems. This causes problems when one wants a simulated computer-graphic object motion (left-handed coordinate system) to directly correspond to the physical sensor motion (right-handed coordinate system). The typical method to transform between left and right coordinate systems is to head the transformation list with a z-reflection matrix—an identity matrix with the 3rd column negated. However this causes two problems. One, it inverts objects in z, and two, the zero rotation position is changed.

To map the naturally right-handed values coming from a three-space sensor to a left-handed coordinate system while preserving motion correspondence, for instance to "fly" an object around, one can treat the effect of sensor motions as if they were in a left-handed system. Translations reported as positive z from the sensor become translations in $-z$. Positive rotations reported in *elevation* and *roll* become negative in the left-handed system. This transformation preserves the physical appearance of the motions across the coordinate systems. Thus, moving the sensor to the left causes motion to the left, and clockwise rotation generates clockwise rotation.

One way to mathematically achieve the transformation is to take position and euler angle readings from the sensor, negate the z translation, and elevation and roll angles, and form a new transformation matrix. Unfortunately rotation angles have a singularity at high elevation (roll is undetermined at elevations of $\pm 90°$) that can cause wild rotations in the matrix. Both the Polhemus (although not when incorporated into the VPL DataGlove unit), and The Bird have alternate methods of reporting the sensor orientation that avoid these problems. The best way to get the sensor data from the Bird is via the following full 3x3 rotation matrix.

$$\begin{bmatrix} \cos(e)\cos(a) & \cos(e)\sin(a) & -\sin(e) \\ -\cos(r)\sin(a)+\sin(r)\sin(e)\cos(a) & \cos(r)\cos(a)+\sin(r)\sin(e)\sin(a) & \sin(r)\cos(e) \\ \sin(r)\sin(a)+\cos(r)\sin(e)\cos(a) & -\sin(r)\cos(a)+\cos(r)\sin(e)\sin(a) & \cos(r)\cos(e) \end{bmatrix} \quad (5)$$

An analysis of this matrix shows that it can be transformed for use in a left-handed coordinate system by negating the elements (0,2), (1,2), (2,0), and (2,1), corresponding to the effect of negating the sines of elevation and roll.

9.6 Hand-local reference frames and transforms

Reference frame

When developing libraries of code for whole-hand input it is useful to have a "standard" reference frame for the hand. The reference frame comes into play when analyzing gestures and when matching hand orientation to controlled object orientation. The convention used in this work for the hand-local reference frame (Figure 24) is centered in the palm with the positive x axis along the extended fingers, the positive y axis to the ulnar side of the right hand and, positive z pointing out the palm.

Hand orientations

Hand orientations can be derived from the three-space sensor matrix. The method is to take a unit vector that lies in the desired direction when the hand is in the zero-rotation state and find its new direction with the rotated hand. The three most useful unit vectors figs/hand-ref-frame.eps

Figure 24: *Hand-local reference frame* The local reference frame used for posture and gesture recognition.

are those that correspond to the direction of the fingers (1, 0, 0), the direction of the palm (0, 0, 1), and the direction of the thumb (0, –1, 0) for the right hand and (0, 1, 0) for the left. These have been termed the *finger direction*, the *palm direction*, and the *thumb direction*, respectively.

Left-right/forward-back One can find the left-right/forward-back orientation (relative to the x-z plane) of a part of the hand by looking to see the rotation angle of the corresponding unit vector $\mathbf{V}$ around the y reference axis. For instance, the palm vector, (0, 0, 1), is used to find the direction of a "waving" gesture (see page 9.7).

$$angle = \tan^{-1}\left(\frac{\mathbf{V}_x}{\mathbf{V}_z}\right) \qquad \begin{array}{rcccll} -\pi/2 & < & angle & < & \pi/2 & towards\ user \\ \pi/2 & < & angle & < & -\pi/2 & away\ from\ user \\ 0 & < & angle & < & \pi & right \\ -\pi & < & angle & < & 0 & left \end{array} \tag{6}$$

Up/down One can find the up-down orientation of a part of the hand (for instance, the thumb [0, –1, 0] to recognize "thumb's up" or "thumb's down") by looking at the angle between it and the y reference axis in the hand-local coordinate system.

$$angle = \cos^{-1}(\mathbf{V}_y) \qquad \begin{array}{rcccll} 0 & < & angle & < & \pi/2 & up \\ \pi/2 & < & angle & < & \pi & down \end{array} \tag{7}$$

Centering and scaling hand translations

As with the default orientations, the default origins for the three-space trackers often are not in a convenient location for whole-hand input use. It is a simple operation to reposition the origin by adding an offset to the x, y, and z translation values in the position matrix. The whole-hand input software library maintains a position vector describing the current center of the whole-hand input space. The library also has an auto-centering function which takes the current sensor values of the three-space tracker position and sets them as the center of the hand space.

Most three-space trackers report position in terms of inches or centimeters. Occasionally it is desirable to scale these values to attenuate or accentuate hand motion. For instance, in the high-level grasp mode for the six-legged walker (page 175) hand motion is greatly attenuated to permit fine control over the position of the object in the walker's grasp. In other cases, the units of the task space may differ greatly from the units of the hand space. It is easy to imagine cases where the task involves large robots or machinery, or, at the other end of the spectrum, small manipulators such as might be used in microsurgery, or circuit repair. Scaling the motion of the hand is a simple matter of applying a scale factor to the x, y, and z translation values in the position matrix. The whole-hand input library maintains a scaling vector for this purpose.

Combining centering and scaling yields a linear relation between the incoming sensor values and the resulting task-space parameters,

$$task_i = scale_i * sensor_i - center_i \qquad i = x, y, z \tag{8}$$

Zeroing and scaling hand rotations

In many situations it is desirable for hand orientations to be relative to a "starting" orientation. This is the case in the orientation experiment (Section 7.2) where the object orientation is relative to the hand position of the "grab." The procedure for doing this involves premultiplying the 3x3 hand-orientation matrix ($\mathbf{M}_h$) by the inverse of the 3x3 matrix of the "starting" orientation ($\mathbf{M}_s^{-1}$). The resulting 3x3 matrix ($\mathbf{M}_r$) will be the relative orientation of the hand from the "starting" orientation. (Because of its equivalence to the inverse for orthogonal 3x3 rotation matrices, the transpose can be used for faster computation.)

$$\mathbf{M}_r = \mathbf{M}_s^{\mathrm{T}} \mathbf{M}_h \tag{9}$$

Scaling hand rotations can be more complex. If the three-space sensor readings are received as three rotation values, e.g., *azimuth*, *elevation*, and *roll*, then it is a simple matter to multiply the rotations by the scale value and assemble the rotation matrix according to (5). If the sensor values come in as a 3x3 matrix, then the process is more complex and involves decomposing the matrix into constituent rotations, scaling the rotations, and recomputing the matrix using the new values. The problem with this method is that the set of rotations that make up an orientation matrix are not unique. The rotations that are extracted by this method will recombine to create the original orientation matrix, however comparing two matrices by this decomposition method may not always give the expected result. The decomposition of the matrices is a straight forward process of solving the nine equations of the 3x3 rotation matrix, $\mathbf{M}$, (5), in three unknowns, a, e, and r (corresponding to rotations about the z, y, and x axes respectively).

$$\begin{aligned}
\text{for } \mathbf{M}_{0,2} \neq \pm 1.0 \qquad a &= \arctan(\mathbf{M}_{0,1}/\mathbf{M}_{0,0}) \\
e &= \arcsin(-\mathbf{M}_{0,2}) \\
r &= \arctan(\mathbf{M}_{1,2}/\mathbf{M}_{2,2}) \\
\\
\text{for } \mathbf{M}_{0,2} = \pm 1.0 \qquad a &= 0.0 \\
e &= \arcsin(-\mathbf{M}_{0,2}) \\
r &= \arctan(\mathbf{M}_{1,0}/\mathbf{M}_{2,0})
\end{aligned} \tag{10}$$

The effectiveness of three-space controls that use scaled rotations is not well known. The technique should be experimented with before committed to an input strategy. As another note, scaling and centering of translations and orientations should be performed separately and the full 4x4 matrix assembled afterwards.

View-independent hand motion

For computer graphic applications, it is important that hand motions indicating directions relative to the screen, maintain their meaning as the screen view changes. For instance, the act of pointing in a direction for an object to move should occur in the coordinate system of the viewer: pointing to the left of the screen, should cause the object to move to the left of the screen. This avoids the problem of reversed control input characteristic of radio-controlled model airplane flight: when the airplane is flying towards the radio controller then the left and right stick motions are reversed.

To perform the transformations, the hand matrix must be first transformed into the coordinate system of the viewing space, and then aligned with the viewpoint and view orientation.

$$\mathbf{M}_{result} = \mathbf{M}_{hand}\ \mathbf{M}_{h-to-v}\ \mathbf{M}_{view} \tag{11}$$

$\mathbf{M}_{h-to-v}$ is the transformation from the whole-hand input testbed standard reference frame to the viewing reference frame as specified by the application being used. In the case of the bolio system this involves a right-to-left-handed coordinate system transform followed by $+90°$ and $-90°$ rotations in the z and x axes respectively.

Fingertip positions

Calculating the position of the fingertips can be done only as accurately as the hand device can record the position of the hand and the angles of the finger joints. For instance, the DataGlove MCP sensors are highly correlated with each other and, without going through the lengthy (and imprecise) process of calibrating for those correlations, the fingertip positions cannot be ascertained with any reasonable degree of accuracy.

Assuming that accurate readings can be achieved, the procedure for determining the position of the fingertips is shown in Figure 25 below.

figs/fingertip-calc.eps

$$\begin{aligned} Pp_z &= Po_z + Lp \, \sin Jp \\ Pm_z &= Pp_z + Lm \, \sin Jm \\ \mathbf{Pd_z} &= Pm_z + Ld \, \sin Jd \end{aligned} \tag{12}$$

$$\begin{aligned} Pp_x &= Po_x + Lp \, \cos Jp \\ Pm_x &= Pp_x + Lm \, \cos Jm \\ \mathbf{Pd_x} &= Pm_x + Ld \, \cos Jd \end{aligned} \tag{13}$$

$$\mathbf{Pd_y} = Po_y + (\sin Ja) \sqrt{\left({Pd_x}^2 + {Pd_z}^2 \right)} \tag{14}$$

Figure 25: *Fingertip position calculation* Fingertip coordinates ($\mathbf{Pd_x}$, $\mathbf{Pd_y}$, $\mathbf{Pd_z}$), calculated from base of finger ($\mathbf{Po_x}$, $\mathbf{Po_y}$, $\mathbf{Po_z}$).

9.7 Gesture recognition

Gesture recognition is distinct from posture recognition in that it requires the recognition of hand shape in both the space and time domains, rather than only the space domain. Stokoe (1960) characterizes signs in American Sign Language with three aspects, *dez, sig*, and *tab*, corresponding to hand shape, hand motion, and body location of the motion (i.e., near the left side of the face). Posture recognition deals with the first of these. Gesture recognition handles the second. Body location can be included in either posture or gesture recognition.

There is virtually nothing in the literature about three-dimensional continuous gesture recognition. Most of the gesture recognition literature has to do with input tablet based recognition of hand-written characters (Martin et al., 1990), or two-dimensional gestures used in text editing (Wolf and Morrel-Samuels, 1987) and 2-D graphics systems (Buxton et al., 1983; Grissom, Carlson, and Perlman, 1989). Characteristicly, these systems depend on explicit specification of the beginning and ending of a gesture, require extensive training, and do not address input spaces of greater complexity than two degrees of freedom.

Gesture recognition for whole-hand input should be able to handle at least the three degrees of freedom of palm translation, if not the six degrees of palm translation and orientation, and the twenty-odd degrees of freedom of finger motion. Unlike tablet-based gestures, there is no natural convention in whole-hand input for signaling the beginning or end of a gesture, such as touching and removing the pen from the tablet. It is desirable that whole-hand input gesture recognition be continuous, requiring no explicit delineation of gestures.

Finally, it is desirable to reduce the training time for any set of gestures as much as possible. Although training times on the order of hours are acceptable for systems that are used heavily by a single person, and training times on the order of minutes for less heavily used systems, very short (seconds) or no training times are ideal.

The work by Fels (1990) used back-propagation neural nets for recognition of simple gestures (various speeds of moving the hand left, right, up, down, in, or out) but required extensive training and explicit specification of the beginning and end of the gesture (page 69). Work by Brooks (1989) used a different form of neural net to recognize simple finger-based gestures such as "closing the hand," "closing the hand leading with the little finger," and "pen grasping" (page 69). This work took a more flexible approach to gesture recognition and used a ten-dimensional hand-space (based on the ten DataGlove finger-flex sensors) as input to the recognition process. Training times were on the order of three to five repetitions of the gesture. Brooks' method performed continuous gesture recognition and did not require explicit specification of beginnings and endings of gestures. Thus, cyclical gestures could be recognized, as well as gestures embedded in other, random, motions.

Rubine (1991) introduced a method of gesture recognition that analyzes continuous features of a gesture path. Interpretation of these features, such as path curvature, bounding box, direction of motion, and so on, can be used as parameters in the recognition of gestures. The advantages of Rubine's method over previous methods is that the features are relatively continuous and simple to compute, so they can be abstracted and interpreted in real-time. He describes the method primarily for two dimensional paths, although it is easily abstracted to more dimensions and to multiple paths.

For the whole-hand input testbed gesture recognition routines and for the prototype applications described in Section 10, Rubine's feature analysis has been extended to three-dimensions and modified to permit continual analysis and recognition without explicit beginning and end points. The feature analysis routines keep a one-dimensional record of values over time for each of the flex sensors and a three-dimensional record of three-space tracker position over time. (At this point, feature analysis of the hand orientation vector is not performed.) As in Rubine's work, the feature set was chosen empirically to be useful for recognition of specific gestures.

The feature recognition routines are open-ended so that addition of new features or modification of existing features can be performed easily.[4] Interpretation of features for gesture recognition was done using an explicit formulation for each gesture, rather than by means of a generic pattern recognition algorithm as is used by Rubine. Advantages of explicit formulation are that gesture recognition routines need no training or samples from individual users. If properly formulated, the routines will work for all users and efficiently use only relevant features. Disadvantages include that the user may need minor training to produce some of the gestures, and new gestures require new formulations.[5] Figure 26

[4]Future work should develop a comprehensive set of features that would allow detection of a majority of useful gestures.

[5]Again, future work should develop more general methods for interpreting gestures. See Section 11.4 for more commentary on this issue.

illustrates the concept of feature analysis.

figs/feature-analysis.eps

Figure 26: *Feature analysis* This figure illustrates the concept of feature analysis for a two-dimensional path. $\mathrm{x}_{i's}$ are sample points along the path.

Gesture recognition algorithms used for the prototype applications combined feature analysis of the three-space tracker and flex sensors with posture recognition and hand orientation information. Initially, a general set of features was implemented based on Rubine's descriptions as an initial base. This was augmented with auxiliary features as required to recognize specific gestures. The initial set was:

- current path-segment vector: $\mathbf{v}_i = (x_i - x_{i-1},\ y_i - y_{i-1},\ z_i - z_{i-1})$
- length of path-segment: $\|\mathbf{v}_i\|$
- normalized path-segment vector: $\bar{\mathbf{v}}_i = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}$
- current speed: $\mathbf{s}_i = \frac{\mathbf{v}_i}{t_i - t_{i-1}}$
- current linear speed: $s_i = \|\mathbf{s}_i\|$
- normalized cross-product: $\mathbf{c}_i = \bar{\mathbf{v}}_{i-1} \times \bar{\mathbf{v}}_i$
- normalized dot-product: $d_i = \bar{\mathbf{v}}_{i-1} \cdot \bar{\mathbf{v}}_i$
- bounding volume: $(x_{max},\ x_{min},\ y_{max},\ y_{min},\ z_{max},\ z_{min})$
- cumulative length: $\sum_{i=0}^{n} \|\mathbf{v}_i\|$

The implementations of specific gestures are described in Sections 10.1 and 10.2 with the exception of "waving," a composite gesture described in the following paragraphs to illustrate explicit gesture recognition techniques.

The "waving" command gesture was used in the six-legged walker application described in Section 10.1. Users wave their fingers (like waving "good-by") in the direction they wish the walker to travel. After a few of the back-and-forth motions, the gesture is recognized and the walker turns in the direction indicated by the palm orientation.

The command is composed of two major parts, the symbolic interpretation of the discrete gesture "waving" and the mapped interpretation of the continuous quantity "palm-direction." The gesture uses the MCP joints of the four fingers and the orientation of the three-space sensor.

1. Confirm that the hand is not closed by using the simple posture recognition of making sure the MCP values are less than 0.8.
2. Look at an auxiliary feature that counts the number of direction switches (local maximum and minimum) in a single valued variable over the last N frames. For instance, the flex-value sequence (.21, .30, .37, .40, .36, .23, .15, .10, .14, .23) contains two direction switches. If this value is less then 5 for any of the four MCP joints over the past N frames then waving is not detected. (N was set to 7 frames at 4 frames/sec update rates.)
3. Check that the range of the waving motion is large enough to avoid catching small random motions of the fingers. This is accomplished by accumulating the path-segment lengths through a linear causal IIR filter,

$$y_{i+1} = kx_i + (k-1)y_i \qquad k > 0.5 \tag{15}$$

   This effectively removes input samples more than N frames old from the accumulated value. If the value is not within an empirically-based range for any one of the four MCP joints then waving is not detected.
4. If all of the above tests indicate "waving," then compute the direction of the palm in the horizontal (x-z) plane as described in (6) on page 157 and return "waving" along with the palm-direction. Otherwise, do not recognize "waving."

10 Demonstrations of Prototype Applications

This section describes demonstrations of three prototype applications that use whole-hand input, and one actual application of whole-hand input in musical performance. The section is included to show how whole-hand input might be used in practice, and how the issues that need to be addressed come into play in the development of an interface. Observations about the effectiveness of the whole-hand input controls in each section are based on formal and informal testing of whole-hand input expert and novice users performing tasks in the applications. In many cases, actual data were collected but either were insufficient for statistically significant results or were not formally analyzed.

The three prototype applications were developed using the whole-hand input testbed with the bolio interactive graphical simulation platform (Section 8). The musical performance application used a 80386-based personal computer for the whole-hand input implementation.

10.1 Robotics: Six-legged Walker/Manipulator

The *six-legged walking robot* is a simulation of mobile legged "vehicle" with a pair of manipulator arms. It was constructed to simulate the kinds of tasks a mobile robot may be required to perform, and to investigate multi-modal use of the hand within a single application. Here it can be used to retrieve and stack blocks. There are several levels of control for the walker, dynamically interchangeable, all operated with whole-hand input control. In addition there are conventional controls that can be used at the varying levels of interaction. All hand positions are described for the right hand.

There are four basic modes of control. Low-level walking, low-level grasping, high-level walking and high-level grasping. In addition to the four modes of control there are three "cameras": a default view, and view from behind the walker's head, and a moving view that tracks the walker and can be adjusted with whole-hand control. Versions of the controls were developed for whole-hand input and conventional device input.

figs/sixleg.ps

Figure 27: *Six-legged walker*

Low-level whole-hand controls

Low-level controls use *direct interpretation* to control the degrees of freedom of the walker. The low-level and high-level whole-hand controls described in this section are summarized in Figure 28.

Low-level walking

In this mode, the user mimics walking with the index and middle fingers. The MCP and PIP joints are linearly mapped to the knee and hip joints respectively of the walker. The walker leg joints follow the finger motions, alternately fixing and releasing from the floor (when one touches, the other releases). The walker body pivots over whichever foot is on the floor, moving forward or backwards as controlled by the user. A tripod gait is maintained by giving the user control over the left and right front legs and slaving the middle and rear pair appropriately. Turning is achieved by turning the hand left or right. There is a twenty degree "dead" zone in the center of the hand orientation. To the left of this, the walker turns left; to the right, the walker turns right. The more left or right the hand is turned from the center, the faster the walker turns.

A good linear relation between finger joints and legs was empirically determined to be

$$\begin{aligned} \phi_{hip} &= 90 * f_{MCP} - 50 \\ \phi_{knee} &= 110 * f_{PIP} \end{aligned} \tag{16}$$

figs/hand-controls.eps

Figure 28: *Whole-hand input controls for six-legged walker*

– where f_j is the normalized value $(0 - 1)$ for a joint j. This gave the hips a range of 50° flexion (forward) to 40° extension, and the knees 0° extension and 110°flexion. Both ranges are close to the joint ranges of the fingers, although a straight hip was mapped to a slightly bent MCP joint.

A detailed evaluation of the finger-to-leg control structure appears in Section 7.1. In summary, this method of control was selected for its naturalness and fine degree of control over leg placement. Leg placement can be important when navigating cluttered, uneven, or discontinuous terrains. Likewise, the turning control is a natural extension of mimicking the walker motions with the hand.

To disengage the hand-to-leg control for mode switching or to rest, the ring PIP joint can be flexed. This *clutch* (as in "automobile clutch") convention is carried through to most walker controls. Extension of the ring PIP resets the "zero-orientation" of the direction control to the current orientation of the hand.

This control mode turned out to be quite successful for short distances and detailed maneuvering. For longer distances there was a trade-off between switching modes to high-level walking (with an admittedly clumsy interface) or risk fatigue by covering the distance "by hand." A problem with the method for turning was that sometimes the walker would unintentionally turn as users forgot to keep their hand oriented steadily. As users gained experience, this effect went away.[1]

Mode-switching

A common set of controls is used to switch between the modes of whole-hand input control. The *clutch* (flexing the ring PIP) must be engaged to enable mode-switching.

- **Control level**: Flexing, extending, and flexing again the thumb IP, cycles from low to medium to high levels of control. This action of flexion, extension, flexion is termed *clicking* the joint and is used throughout the walker interface.
- **Camera**: Clicking the index PIP joint cycles the cameras from default, to head-view, to the movable-camera.

The ring PIP was selected for the *clutch* because it is one of the least dexterous of the finger joints and so works well as a binary switch, and is independent from at least the index and middle fingers used for most of the other elements of the control tasks. Some users had difficulty moving this joint independently of the ring MCP, pinkie MCP, and pinkie PIP, and some users could only perform the action well on their dominant hand. However, most users found that with a little practice they could use the joint as a binary switch. The index and thumb "click" actions were chosen for their independence from the ring finger clutch, similarity to "flicking a switch" or pushing a button, and ease of use. Users had little difficulty using these hand actions for mode control.

Low-level grasping/manipulating

Grasping is distinct from walking and only one of the two can be in effect at a time. In grasping mode, the walker's manipulators, attached to the head can be used to retrieve and manipulate an object. An object only can be released in the grasping mode, so that while walking, it is carried in the manipulator arms. In the low-level grasping mode the thumb and index joints control the left and right shoulder and elbow joints in a master-

[1]The phenomena of forgetting to maintain little used, but in-use, degrees of freedom is common for novice whole-hand input users. This was observed both here and in the mapped controls for the crane application (see page 180).

slave relationship. A maximum angular velocity can be set for the arm joints. If this is low enough, then the operator's fingers control joint goals which are achieved by the arms over time. This approximates physical robotic motion constraints and introduces lag in the walker's slaving to mastered positions, without introducing lag in the walker's response to changing goal positions.

Moving the head and arms

- A *wrist flick* toggles between walking and grasping modes. This is performed by a roll of the wrist of approximately 50° in one direction and then back again while the ring PIP is bent (to clutch walking or grasping controls). The angle of the wrist flick is measured by computing the rotation matrix $\mathbf{R}$ that expresses the change in rotation of the hand between the last two hand sensor samples $\mathbf{H}_i$ and $\mathbf{H}_{i-1}$

$$\mathbf{R} = \mathbf{H}_i \mathbf{H}_{i-1}^{\mathrm{T}} \tag{17}$$

  and then extracting the r rotation using (10) on page 159. Recall that the r variable also represents rotation around the x axis and corresponds to wrist roll in the whole-hand input standard hand reference frame. When the walker is switched back into walking mode, the head centers itself left-to-right, but maintains its elevation (a convenient feature so that the user can set the view for close in or far ahead). When in grasping mode, the head appears orange in color; when in walking mode the head appears white. This helps cue the operator to which mode is currently in effect.

- Thumb MCP and IP, and index MCP and PIP control the elbows and shoulders of the left and right arms respectively. A good relationship between finger joint and robot joint was empirically determined to be

$$\begin{aligned} \phi_{shoulder} &= 90 * f_{MCP} - 40 \\ \phi_{elbow} &= 90 * f_{PIP} \end{aligned} \tag{18}$$

  – where MCP is the index MCP or thumb MP joint and PIP is the index PIP or thumb IP joint.

- Roll of the palm rotates the arms up or down at the shoulder joints. When the palm is rolled left, the left arm drops and the right arm raises. When the palm is rolled right, the right arm drops and the left arm raises. The neutral position is with the palm facing left. Limits of 20° up or down were placed on this motion as arbitrary "robot-like" joint limits.

- Azimuth of the finger direction controls the left-right orientation of the head. This motion was also arbitrarily limited to 20° left or right.
- Elevation of the finger direction controls the elevation (up-down orientation) of the head (clamped to 20° up and 90° down— 90° was needed to retrieve and place objects directly below the walker).
- Flexion of the PIP of the ring finger (the *clutch*) disengages the fingers' effect on the manipulator arms and the hand's effect on the head.

These controls were selected because of the natural mapping between finger joints and walker arm joints. Following this model, the motion of the hand naturally maps to the motion of the head. This provides nearly independent control over six degrees of freedom in a natural manner. Users found this model of moving the arms both simple and intuitive.

Manipulating objects

One of the goals of this application prototype is to investigate modes of controlling robotic manipulators with whole-hand input. The two arms of the walker can be considered as two manipulator arms or as an abstraction of two dexterous fingers of a robot gripper. Attachment to the object was treated as a separate problem and simulated by making the object simply stick to the walker hands. The control desired was one in which an object could be manipulated by the walker to any orientation or position. Some liberties were taken, such as the object not moving when released, however, the control structure that was developed allowed arbitrary orientation and positioning the object without requiring the object to be released.

The implemented controls were chosen as one method of many studied. They represent a compromise between the computational power of the implementation platform and the goal of providing an accurate simulation of robotic control.

When a hand touches an object in grasping mode, it sticks to the object. An object can be grasped in one or two of the walker's arms (hands). The behavior of each hand in relation to the object is slightly different depending on whether the hand was the first or second to make contact with the object. To tell the two apart, the first hand to grab is colored red, the second is colored green. The hands are white when not grasping the object.

- When an object is grasped by one hand, then the object acts as a kinematic child of the hand at the point the hand touched the object.

- When an object is grasped by two hands, the object moves so as to follow both hands, maintaining both points of contact. Since this causes a closed kinematic loop, the first hand to touch follows the motion of the controlling finger. The object moves with this hand, but rotates around its point of contact to align the second point of contact with the position of the second hand (i.e., the vector formed by the touch points on the object is aligned with the vector formed by the two hands). The position of the second hand is subsequently moved by an inverse kinematic routine to touch the object at the second point of contact. The effect is that the two controlling fingers can control the object as if the object were actually with in the user's grasp.

- Release of the object is accomplished by a flexion of the pinkie PIP joint. Multiple flexes of the pinkie while the clutch (ring PIP flexion) is engaged perform different patterns of release: one flex, release the first hand; two flexes, release the second hand; three flexes, attach the object to the second hand as if it were the first to grab; subsequent flexes toggle this last state, alternately releasing and attaching the second hand.

  The pinkie was chosen since its motion is independent enough from the thumb and index finger to not disturb the position of those fingers when the object is released. Hand motion, such as wrist-flick was considered, but tended to disturb the thumb too much (the index lies along the axes of rotation and seems to be less disturbed by wrist roll than the thumb lying perpendicular to the rotation axes[2]). Some subjects found it difficult to move the pinkie without moving the ring finger. For this reason, the pinkie release control does not require the ring finger to be opened. Few subjects had problems moving the pinkie when the ring finger was closed.

The range of proficiency using this method of control varied greatly over tested subjects depending on their dexterity (as evidenced by performance in other whole-hand input tasks), and familiarity with whole-hand input, kinematics, and on-screen object manipulation. Those that could do well had few problems grabbing, re-orienting, and placing objects. Others had some problems, but got better as they gained experience.

The main difficulty seemed not to be with using the hand to control the walker arms,

[2]This and similar effects need further study and would be excellent topics for future research.

but with the kinematic differences between the fingers and the arms. The effects caused by the walker arms being constrained by a kinematic loop with the object, and that the user's fingers had no such constraints, hindered many of the users. Those with a strong understanding of what was going on had less of a problem than those who knew little about kinematics. The most skilled subjects were able to manipulate objects with this mode as well or better than with the high-level manipulation mode.

This range of experiences across users indicates that there is much that can be done towards improving the understanding of whole-hand to task mapping, particularly in the area of direct and mapped interpretations in robotic control (see Section 11.8).

High-level whole-hand controls

High-level controls allow the user to act in a supervisory mode, directing pre-programmed behaviors of the walker.

High-level walking

The walker has an automatic gait controller that allows it to walk on level ground using a tripod gait. The high-level walking mode has four commands: speed up, slow down, stop, or turn.

- **Speed up**: A flexed index and middle finger with the thumb held up accelerates the walker in the forward direction (and decelerates the walker in the reverse direction).
- **Slow down**: A flexed index and middle finger with the thumb held down decelerates the walker in the forward direction (and accelerates the walker in the reverse direction).
- **Stop**: Flexion of the ring PIP (the *clutch*) stops the walker (as well as enabling the use of mode switching). Often users simply made a fist to stop the walker.
- **Turn**: Waving the fingers (at the MCP joints) three times in moderate succession (not too fast, not too slow) signals the walker to turn in the direction the palm faces on the third or fourth wave. The direction is view independent, that is to say, a wave towards the left of the screen, causes the walker to move to the left of the screen, regardless of the camera view. With the default view, a left wave will turn to the

left of the work area. With the head view, the walker will turn towards the left of the body. (Section 9.7 describes the implementation of the waving gesture.)

The waving was initially chosen as a natural interface to commanding direction. Subjects found that although it was natural, intuitive, and even fun, it did not provide enough precision for a guiding task. The delay inherent in executing the waving motion and the indeterminate timing of the gesture recognition prevented precision timing of turns. People's apparent lack of ability to judge precisely the absolute direction of the hand without continuous feedback, hindered accurate orientation of the walker once the finger motions were recognized. A better strategy would incorporate continuous direction control, or a direction indicator that continuously displays the goal direction while the user orients the hand in preparation for the turn gesture.

The commands for speeding up, slowing down, and stopping were chosen for their simple mnemonic value and proved adequate to the task. Users expressed an interest in being able to control the setting of the acceleration so that if they had a long distance to walk they could get going more quickly. However, this feature was traded off with simplicity in the control interface. With more thought and another iteration in the design method, a solution could be found.

High-level grasping

In the high-level grasping mode, automatic routines control most of the walker's behavior. The user can command the walker's arms to reach towards and grab an object, to open and release an object, or to manipulate a grasped object. The head is kept in a neutral position, facing forward. A maximum angular velocity can be set for the arm joints with similar effects as in the low-level control mode.

- **Grasping**: Closing the hand (a fist posture) causes the walker's arms to reach for the nearest object. If contact is made, then the hands are brought into a neutral forward position, bringing the object with them.
- **Releasing**: Opening the hand (an open-palm posture) causes the walker's hands to spread apart, releasing the object.
- **Manipulation**: Forming a fist "grabs" the object. Subsequent rotation and translation of the hand rotates and translates the object relative to the hand's position

and orientation when the fist posture was recognized (translations are attenuated so that the object does not translate beyond the grasp space easily).

What actually happens is that hand position determines a desired object position. New arm joint settings are calculated to move the object as close to that position as the arm configuration space will allow. These new positions are set as goals for the arm joints. The hands then move to their new positions (at a speed determined by the maximum joint velocity) carrying the object with them. Thus, the user is not directly controlling the object, and cannot pull or rotate the object out of the walker's achievable grasp space. This process happens every frame-time (about 250ms) so the user is in a tight loop with the walker's arms. This control mode was used for the orientation experiment described in Section 7.2.

The grasp and release commands were chosen for their simplicity and mnemonic value. Users found the commands very useful for quickly obtaining and releasing objects. However, the grabbing routine was not very sophisticated and often chose an awkward grasp with the hands too close together for precise manipulation of the object. Users often grabbed the object in the high-level mode and then, if the hand's attachment points were not optimal, dropped down to the low-level to reset the points of grasp. A detailed analysis of the high-level object manipulation controls is described in Section 7.2.

As a final note, changing the maximum joint velocity of the arms had a significant effect on the performance of the task. Users found that some damping in the arms prevented overshooting their goal orientation. Too much damping made the system too slow, while too little damping led to instabilities in the control loop.

Camera controls

Of the three camera views, one is movable using whole-hand input. When using the movable view the formation of a fist (with a thumb in the air to distinguish the posture from the "grasping mode" fist) will "grab" the viewpoint. Subsequent movement drags the camera with the hand. The movable camera's viewpoint stays centered on a spot just in front of the walker's head. Opening the hand releases the camera.

This feature was not tested by very many people and is one of many possible schemes for controlling a camera with whole-hand input. It was used successfully by the author quite frequently to set camera views while testing versions of the walker, and for arranging screen snapshots for documentation.

Conventional controls

Sets of conventional controls were developed to achieve the same functionality as the whole-hand controls. The conventional interface schemes use nine dials together with a box of 32 buttons, or a Spaceball. These were chosen for their availability and as representative of the kinds of controls most used in robotic interfaces. (A review of the robotics literature suggests that although a mouse is often used in workstation interfaces, it is used infrequently in robotic control.)

The dial box and button box are used together to control the walker as shown in the left and center columns of Figure 29. The Spaceball is used to control the walker as shown on the right of Figure 29. Note that low-level controls were not implemented for the Spaceball. The degrees of freedom of the device are too coupled to control independent degrees of freedom in the simulation. Decoupling schemes that permit the control of one or two degrees of freedom to be controlled at a time tend to become baroque and confusing. Also note that camera control was not implemented for dials or Spaceball because this feature was not part of the control mode experiments. It would be a simple matter to extend the devices to control the view as well.

figs/conventional-ctrls.eps

Figure 29: *Conventional controls for six-legged walker* On the left appear the dial controls for the four modes of walker control. In the center are the button controls used in conjunction with the dials. On the right are the Spaceball controls.

10. Demonstrations of Prototype Applications

10.2 Construction Crane

The construction crane simulation was developed to examine the use of gestures in an application and to demonstrate the possible use of whole-hand input in the construction industry as described in Section 3.2.

figs/crane.eps

Figure 30: *Construction crane simulation* The simulated crane environment.

A simple representation of a construction crane was modeled, based on dimensions of drawings of latticed boom cranes in Shapiro and Shapiro (1988). The simulated crane can move forward, backward, or turn by rotating the left and right treads. The cab rotates on the base, the boom tilts up or down, and the hoist cable reels in or out. The angle of the jib can be modified, but does not change in the course of a task, as is the case for real cranes. The swing of the hook is dynamically simulated as a pendulum bob strung from the tip of the jib.

The task of picking up an object and placing it at another location was used to experiment with the crane controls. Whole-hand input was created for three of the crane's degrees of freedom: the swivel of the cab on the base (referred to as *swing*), the angle of the boom (*boom*), and the height of the hoist (*hoist*). The movement of the treads was disabled in user experiments to simplify the interface.

Two separate methods of whole-hand input were developed for the crane. The first relied on conventional crane operator hand signs, and the second on finger joint flexion. The hand signs were chosen to see if the conventional signs could be recognized by the computer to control the crane. However, these signs evolved to be visible from a distance and are not the most efficient method of crane control. The second method was implemented as a more efficient method of crane control and to examine the ability to coordinate the crane's three degrees of freedom with the hand. Conventional controls using three dials were implemented for comparison to the whole-hand input methods. Several test subjects were run on the simulated pick up and place task. The results were statistically inconclusive and are not included in this document, although anecdotal results were informative, as is reported later in this section.

Gestural controls

Gestural controls for the crane consist of four gestures, each with two directional variations. Three gestures control three degrees of freedom of the crane: swing, boom, and hoist. The fourth gesture coordinates the boom and hoist so that the hook (or *load*) is brought towards or away from the crane but maintained at the same height.

Swing controls the swivel of the cab on the base and thus the side to side travel of the load. The gesture recognition is implemented by looking at the finger direction vector in the horizontal (x-z) plane (page 157) and the hand posture. If the hand posture is a fist with the index extended and the finger direction vector is pointing to the left of the user (angle close to $\pi/2$), then the crane swivels to the left. If the pointing posture is recognized with the hand pointing to the right (angle close to $-\pi/2$), then the crane swivels to the right. Back and forth motion of the hand in the direction of pointing accelerates the motion of the crane. The speed of the hand is determined from the *current speed* variable of the feature analysis of the hand's motion (page 164).

figs/crane-gest.eps

Figure 31: *Construction crane signals* These are four of the hand signals implemented with whole-hand input for use with the simulated construction crane. The top row shows the primary signal, while the lower row shows the signal for the reverse direction. The illustrations are reproduced from illustrations of common construction hand signs sent to the author by a local heavy equipment supplier, MARR Equipment Corporation (Boston, MA).

Boom controls the angle of the boom. The gesture for boom control is recognized by identifying a fist posture with the thumb extended and by examining the up-down orientation of the thumb direction vector (page 157). If the vector points up (angle close to $\pi/4$) then the boom moves up. If the angle points down (angle close to $-\pi/4$) then the boom moves down. Up and down hand motion accelerates the boom motion.

Hoist controls the up and down motion of the load. This gesture is a combination of the pointing posture (fist with index extended), the finger direction vector of the hand pointing up or down, and a circling of the hand in the horizontal (x-z) plane. The circling of the hand is determined from the normalized cross-product of the feature analysis of the hand motion. The cross-product operation produces a vector perpendicular to the original vectors (consecutive segments of the motion path) and of a length proportional to the sine of the angle between the original vectors. If the motion of the hand is clockwise, then the cross-product vector will point in one direction. If the motion of the hand is counterclockwise, then the vector will point in the other direction. Motion in the x-z plane will produce cross-products pointing in the y or $-y$ direction. Since the vectors are all normalized, the recognition algorithm needs only to see if the y component of the cross product is close to $+1.0$ or -1.0 to determine if the motion is in the x-z plane. The magnitude of the vector indicates the size of the circling. The smaller the cross-product, the larger the circle.

figs/circling.eps

Figure 32: *The circling gesture* The circling gesture looks for a fist with the index finger extended up or down and with a cross-product close to vertical (indicating horizontal circling).

The recognition routine ensures that the cross-product is consistent in both size and direction for at least three samples before signaling recognition. Any change in this consistency immediately cancels recognition of the gesture. As with the other gestures, the linear speed of the hand determines the speed of the hoist. A speed threshold is set for initial recognition of the gesture to avoid false recognition on random motion. Once the circling motion crosses this threshold and is recognized, then the speed can be reduced below the threshold, slowing down the hoist without losing the gesture. If the index finger is pointing up, then the load is hoisted. If the index finger is pointing down, then the load is lowered.

10. Demonstrations of Prototype Applications

The direction of spin (clockwise or counterclockwise) is ignored for this control.

Load in or out coordinates the raising (or lowering) of the boom with the lowering (or raising) of the hoist to move the load in and towards (or out and away from) the crane cab while maintaining the height of the load. The gesture for this command is recognized by a similar method to the boom gesture. If the hand is positioned with the thumb extended and pointing up or down, then the boom is moved in the direction indicated by the thumb. If the fingers are moving in addition, then the hoist is moved simultaneously to keep the height of the load constant. The speed of the boom and hoist are controlled by the speed of the finger joint rotations. A threshold on finger movement prevents sensor noise, finger tremor, or small, unintentional motions from triggering hoist motion.

Coordinated finger flex controls

In this mode of whole-hand input control, three of the crane motions—swing, load in or out, and hoist—are rate-controlled by the three flexion angles of the thumb IP joint, the index PIP joint, and the middle PIP respectively (Figure 33). These joints were picked as the most independent and dexterous of the hand. Extension of the thumb IP joint beyond the normalized value of 0.3 ($flex_i \leq 0.3$) causes a left swing of the crane, while flexion above 0.7 ($flex_i \geq 0.7$) causes a right swing. Extension of the index PIP below 0.3 raises the hoist, flexion above 0.7 lowers the hoist. Extension and flexion of the middle PIP below 0.3 and above 0.7 raise or lower the boom. The center zone is left as a "dead zone" to make it easier for the user to stop the motion. Deviation of the flex below 0.3 or above 0.7 controls the rate at which the crane moved.

Conventional controls

Four dials were used for the four crane controls: one for swing, one for boom, one for hoist, and one for load in or out. The dial values were scaled to optimize the trade-offs between precision and speed.

figs/crane-coord.eps

Figure 33: *Coordinated crane controls* Shows the three crane motions controlled by the thumb IP joint, and the index and middle PIP joints.

User performance

In a formal experiment, five subjects used the gesture, finger flex, and conventional device controls to manipulate the crane to pick up a ball, carry it over a wall, and place it in a wire frame. Each subject performed the task four times (with four different ball, wall, and frame configurations) for each of the control methods. The time from the onset of control to placement of the ball in the frame was recorded for each trial. The experiment was discontinued when preliminary analysis of the first five subjects' data indicated that the experimental controls were insufficient and that other tests should be devised. (This led to the experiments reported in Section 7.) The main difficulties were that the controls were not properly optimized for the task. Performance differences were not due to the use of one device or another, but due to how the control structures had been set up. For instance, the gain of the dials did not match the gain of the flex controls.

Nevertheless, informal results based on the data from the five subjects, observations of their performance, and interviews after the trials, were informative about the use of whole-hand input. The subjects found the gestural control the most natural and easiest to learn and remember. However, they found that gestures were not as responsive as the dials (there was a slight lag time) and lacked the sensitivity needed for fine control in parts of the task.

In contrast, the dials were perceived as being responsive and precise, but confusing in terms of which direction of rotation controlled which direction of crane motion. The subjects made many more directional mistakes with the dials than with either of the whole-hand input methods.

The aspect of the gestures that required continual hand motion for rapid crane motion bothered most of the subjects when they were trying to move the crane through large angles (the crane joints had low maximum speeds to better simulate the behavior of real cranes). They would have preferred to be able to set the speed with the gesture, rest while the crane moved, and then use another gesture to stop the motion.

Subjects found the coordinated flex controls more difficult to master than the gestures or the dials, but liked the ability to coordinate all three degrees of freedom. They found that they tended to overshoot their mark more often with this type of control. The cause seemed to be a combination of the absence of tactile feedback from the whole-hand interface device, and the poor quality of human finger absolute position sense (conjectured to be between two and three bits[3]). For similar reasons, subjects found it difficult to the find the center "dead zone," even though it was the center 40% of the range of finger motion.

Another observed phenomena was finger drift. As subjects focused concentration on one or two degrees of freedom, the unattended finger(s) would begin to drift out of position. This caused unintentional and unexpected motion in the crane, usually resulting in confusion and loss of overall control due to inappropriate recovery efforts. As subjects gained experience they were better able to control this effect, but it did not seem to go away completely. This indicates a strong need for some form of tactile feedback in the whole-hand interface (even as simple as a detent at the "center" position) or passive constraints, such as joint damping, to prevent finger drift.

[3]Personal communication with Hong Tan, an MIT PhD student researching passive hand motion sensibility.

Fatigue was not a problem with any of the methods, except with the coordinated finger flex where subjects found it tiring to freeze a joint in the "neutral" position. The inability to disengage from the finger-flex control was also tiring in the long run. However, a "clutch" was not included in the experimental interface because of the relatively short duration of the tests.

During the course of this experiment, a nearby construction project provided an opportunity to interview skilled crane operators and construction workers. They indicated that learning to control cranes properly was a skilled task, requiring a long apprenticeship. Observations of the cranes in action indicated that the tasks require fine, coordinated control.

One operator mentioned that there have been attempts to control heavy equipment remotely. The problem, he said, was similar to that found in remote (model) aircraft flight; the direction of the controls changes as the equipment faces towards or away from the operator. With practice, this problem can be overcome, or eliminated by an intermediary computing device. Several of the construction workers were invited to try the simulation. They found no difficulty using the system, and one said half-jokingly, "It responds better than some crane operators I've worked with."

10.3 Expressive Puppet

The expressive puppet was originally developed as a simple space-station repair robot that could navigate the exteriors of orbiting structures under direct human control. The robot would walk along the outside of the structure using standard footholds as suggested by Minsky (1990). The use of whole-hand input would allow dexterous foot placement on the struts and surfaces of the structures, as well as manipulations of tools in the robot's hands.

Once the robot was modeled and put under direct whole-hand input control, its expressive capabilities dominated the demonstration. Before long, it became more interesting as a puppet, and robot manipulation was relegated to the more sophisticated, but less animated six-legged walker described in Section 10.1.

figs/puppet.eps

Figure 34: *Expressive puppet*

The puppet is pictured in Figure 34. It consists of an octagonal body, with two arms, two legs, and a conical nose (to tell which way is forward). The feet are "sticky" and alternately attach to any surface in the simulated environment. When one foot touches a surface, the other releases. The puppet is made to walk by using the index and middle fingers to mimic the walking motion exactly as is done with the six-legged walker. However, the puppet is not "stable" the way the walker is, and rotates around whatever leg is on the ground. The rotation follows the orientation of the hand. Rotation of the hand left or right rolls the puppet left or right. Rotation of the hand forward or back, or up or down, causes similar rotations of the puppet (all around the planted foot). The puppet's arms are controlled by the thumb and pinkie similarly to the feet. The lower (proximal) joints control the shoulders, and the upper (distal) joints control the elbows.

To accommodate the ability of the puppet to walk in any direction and on the sides, tops, and bottoms of objects, without contorting the hand and arm, the user must be able reorient the hand-to-puppet rotation mapping. This is accomplished by flexing the ring finger which freezes the puppet (the same "clutch" as used in the six-legged walker, and for the same rationale), rotating the hand to a comfortable position, and opening the finger. When the finger is opened, the new orientation of the hand is mapped to the current orientation of the puppet. Subsequent motion is relative to this position.

Over the course of the first few days of using the puppet, the author became proficient at walking it around the environment, gesturing with the hands, and bringing "life" to the character. Others found that they could control the character moderately well with a few minutes of practice. The method of controlling the orientation of the character has its problems. Control works best when the hand is oriented the same as the character. As the orientation of the hand deviates from the orientation of the character, by repeated "clutching" and "unclutching," control becomes more difficult. This is, again, similar to the classic problem of remote control flight. At orientations not parallel to the controller, left and right responses (and in this case, up and down, and forward and back) are rotated or completely opposite to the control's left and right.

Strangely enough, people find that walking backwards is often easier than walking forwards. This seems to be a consequence of the way in which the fingers are coordinated for the walking motion, and how people approach the walking task. Walking forwards with the fingers is mistakenly perceived as a simple task and people tend to perform it too quickly for their level of experience. The most common problem is failure to raise the fingers high enough for the feet to clear the ground on the forward swing, “tripping” the puppet. Walking backwards, however, requires more thought into the mechanics of the action. People perform the motion more slowly and with greater care, leading to better results.

The quick response of the puppet to every nuance of the user’s hand motions gives the puppet a life-like quality. The jitter of the Polhemus sensor on the DataGlove puts the puppet in constant motion adding to the impression of life. Long arms and legs accentuate the moving parts of the character, lending grace to all but the clumsiest of actions. Unskilled users tend to over-correct rotations giving the puppet a drunken appearance. More skilled users can keep it “sober,” except for the occasional blunder. There has not been the opportunity to extensively train a person on the system nor to work with a trained puppeteer.

10.4 Musical performance: Bug-Mudra

In September, 1989, Media-Lab composer Tod Machover decided to try to include some form of whole-hand input in the performance of a piece he had been commissioned to write for the 50th anniversary of Nippon University. Earlier that summer, a visiting student under the direction of Machover had explored the use of the VPL DataGlove for controlling MIDI parameters in real-time (Gialanze, 1989). The results had been promising—it was possible to control parameters in an interesting way—however, the three-space sensor (Polhemus) lag and DataGlove imprecision prevented the consistent and fine control needed for a virtuosic musician.

Since that summer's project, the Media-Lab's Computer Graphics and Animation Group had acquired an Exos Dexterous HandMaster (DHM) which exhibited increased finger-flex precision over the DataGlove as well as having a 100 hertz update rate. Machover planned to use this for his new piece. (In the course of the following months an electronic mailing list was set up for the people working on the project. Since the piece was characterized by its use of whole hand input, the mailing list was called *bug-mudra*. *Bug* for bug-X, a common computer convention for where to send error (bug) reports about software development on project X, and *mudra* for the stylized hand-motions of classical Indian dance. The composer liked this name and adopted it for the final piece.)

The original plan was for the percussionist (playing a vibraphone-like MIDI *hyperinstrument* with four mallets) to wear the DHM and be able to modulate his performance by altering the shape of his hand. Unfortunately the DHM geometrically, as well as dynamically, interfered with the four-mallet method required to play the instrument.[4] The linkages on the DHM interfered with the loose, smooth motion of the mallet sticks, while the mass of the DHM changed the inertial qualities of the performer's arm-hand system. In addition, it was found that the grip necessitated by the four-mallet method precluded modification of the shape of the hand to control extra performance parameters.

It was then determined that the conductor, Machover in this case, would wear the DHM on his left hand and be able to shape and mix timbres with it, similarly to the traditional role of the conductor's left hand. It was felt that whole-hand input was particularly well

[4] In the four-mallet method, the percussionist holds one mallet between the index and thumb with the stick across the palm, and a second mallet between the proximal phalanges of the index and middle finger. The other hand holds two more mallets in a similar fashion.

suited to this application since it permits the smooth, flowing motions desirable for freely shaping sound. Most instruments have discrete actions, such as the piano, or continuous actions with reference to a physical object, such as the cello or violin. MIDI controllers for these classes of instruments exist. The voice has a free continuous range but also has certain dynamic constraints and is difficult to measure. Conducting, however, can be fluid and unconstrained. The orchestra serves as an instrument whose sound is shaped by conductor's motions. This was the effect that Machover hoped to create with the computer through the use of whole-hand input.

The setup is shown in Figure 35. Two guitarists and one percussionist generate acoustic and MIDI signals read by a Macintosh IIci running *Hyperlisp*, a real-time MIDI extension to Macintosh Common Lisp developed by Joe Chung and others in Electronic Music and Cognition Group at the MIT Media Lab. Based on SMPTE timing signals on audio tape and the dynamics of the performer's playing, the master *Hyperlisp* system generates MIDI events for 16 Yamaha TX-816 synthesizers, Kurzweil PX-100 and Roland GR50 sampler/synthesizers, and three Alesis HR-16 drum machines.

The sixteen TX-816 synthesizers are channeled through sixteen channels of two Yamaha DMP-7 computer controlled mixing boards. Input from the DHM is processed first by whole-hand input software (similar to that described in Section 9) running on a 25MHz 80386-based personal computer, then sent via a serial line to a Macintosh IIcx also running *Hyperlisp*. This second *Hyperlisp* system, using cues from the master *Hyperlisp* system running on the Mac IIci, sends commands to the DMP-7s modulating the signals generated by the synthesizers. Using the DHM, the conductor has control over parameters that affect the settings of the mixing board, raising or lowering levels of select portions of the acoustic signals of the performance.

The requirements of the task were that the conductor have tight control over several parameters simultaneously. This meant a high update rate and fast response to the hand input. The degree of accuracy or precision was set at eight bits. Control needed to be continuous, repeatable, and easily mastered since there were only a few hours of practice available. It also had to be somewhat intuitive and natural since the conductor had to split his concentration between the hand controls and conducting the three musicians through some very difficult passages of music. The piece was fifteen minutes long, which defined the endurance required.

figs/mudra-system.eps

Figure 35: *Hyperinstrument setup for* Bug-Mudra

The task for *Bug-Mudra* was continuous and non-kinematic and fell into the *continuous-mapped* category of the whole-hand input taxonomy.

Several passes were made at whole-hand input methods until the final solution was reached. Initially a small set of gestures were defined to switch between several modes of use, but these were abandoned when it became apparent that the gesture recognition software was not robust enough to guarantee immediate recognition of hand postures one hundred percent of the time, and that without positive feedback, modes would be too hard to keep track of or verify in the heat of the performance. It was determined that the controls would be single-moded along the nature of virtual sliders, mastering the actual sliders of the DMP-7. This would provide a cognitively simple mapping from hand-space to control-space.

The next choice was between rate control and position control. Position control was the preferred option because, in practice, it was difficult to use rate control for more than one level simultaneously, especially when there was no visual feedback of the levels being mixed. With position control, every position of the fingers corresponded to a position of the mix levels.

It was decided that four parameters would provide sufficient control where each parameter controlled a different group of DMP-7 sliders at different times during the performance. The grouping was controlled by the Mixer Mac, taking its cues from the Master Computer and the SMPTE time code. At times the grouping was such that the conductor controlled the relative volumes of each of the instruments, at other times different tone qualities were assigned to different groups, so that one parameter controlled the level of the purest tones, while another parameter controlled more complex tones, and a third affected the very complex tones. Changing the mix of these changed the "color" of the entire performance.

Initially it was thought that the flex of the four MCP joints could be used to "ratchet" the values of the DMP-7 sliders. "Ratcheting" is the process of moving a finger to its limit, "uncoupling" the control by flexing the thumb, resetting the finger, and "recoupling" the control by opening the thumb. In this way the parameters could be modified in several stages of fine control.

A problem with the "ratchet" method is that, although it works for one or two finger controls, it becomes overly complex for four fingers, simultaneously traveling in some combination of two directions, all controlled by a single "clutch." Working the DMP-7 parameters this way could only be done slowly with great concentration. The "ratchet" was abandoned in favor of a direct mapping from finger flex to parameter value. In conversations with the conductor it was felt that the best arrangement for him, in terms of dexterity and ease of use, was to use the flexions of the thumb IP joint and the MCP joints of the first three fingers as the controls for the DMP-7 parameters. An open finger meant setting the parameter to the top of its range, and a closed finger meant setting a parameter to the bottom of it range.

Initially, linear mapping was used between the top and bottom of the range. However in rehearsals the conductor expressed the wish that the controls be more sensitive at the ends of the range at the cost of midrange control. A series of exponential mappings was generated as lookup tables converting between DHM sensor readings and output parameter values. The function used was an exponential, $y = e^{Ax}$ with $0 \leq x \leq 0.5$, normalized to $0.0 - 0.5$ and reflected about the point $(0.5, 0.5)$. Varying A changes the flatness, or sensitivity, of the curve at the extremes, and the steepness, or lack of sensitivity, in the middle. Values near the two extremes were clamped to 0 and 1 so that the full range of control was available even if the calibration of the DHM was a little off. A series of values of A was tested by the conductor who chose one based on the feel of the control it gave him. The mapping table he chose is shown in Figure 36.

figs/todcurve.eps

Figure 36: *Exponential mapping used in* Bug-Mudra This curve shows the mapping from flex sensor input to parameter output for the four sensors used in *Bug-Mudra*. Note the shoulders at the end of the range. This allowed the conductor to control the full range of the parameter even if the DHM calibration was a little off.

In summary, Machover set up the controls so that three of the four parameters were controlled by his (the conductor's) index, middle, and ring fingers and affected the settings of different combinations of sliders on the DMP-7 depending on the section of the score (as determined by the reference audio tape and the *Hyperlisp* system), while his thumb controlled the fourth parameter affecting the left-right pan of other acoustic channels. Conceptually, Machover arranged the controls so that during portions of the piece, each finger controlled a different instrument, while at other times the index finger controlled the purest tones, the middle finger controlled the intermediate tones, and the ring finger controlled the most complex tones. He cognitively abstracted the whole process using hand shape as a cue to the timbre for the overall sound at any one point.

One of the problems encountered in the use of the hand was the lack of reference points for precise control. The fingers have poor sensation of absolute position, and the changing sound of the music was the only reliable feedback. As was reasoned in the crane application (Section 10.2, page 185), some form of kinesthetic feedback would improve the interface.

As a debugging and status monitoring aid, the PC was programmed to show a graphic display of the hand, the values of the four parameters, and the continuous stream of bytes being sent out to the *Hyperlisp* system. This turned out to be invaluable during performances for several reasons.

One was that at the beginning of each performance a short calibration process was necessary. As each instrument had a short "plug-in" and tuning session, so did the DHM. The conductor walked on-stage, took his bow, plugged the DHM into a cable on the floor, turned towards the DHM computer operator, and opened and closed his hand several times. The whole-hand input code on the PC took this as calibration data, finding the minimum and maximum values, and set the appropriate parameters in the sensor-to-flex functions. The graphic display was activated and the calibration tested by having the conductor open and close his hand again. If the hand on the screen opened and closed, and the output parameters reached their maximum and minimum correctly, then the calibration was complete. If the full parameter range could not be achieved, then the calibration was repeated. This whole process took between ten and fifteen seconds, and guaranteed that the conductor would have the proper control of the DMP-7 parameters during the performance. It served as a nice introduction to the technology for the audience as well.

During the performance the display allowed monitoring of the hand and of the data passing out to the *Hyperlisp* system. If anything went wrong, the source could be quickly isolated. If the problem was with the DHM then little could be done, since the piece was in progress. However if the link to the computer or between the computers was accidentally severed then the problem could be identified and quickly corrected. Fortunately, there were no problems in the course of the fourteen or fifteen performances given over an eighteen month period.

Finally, the on-screen graphics served as a useful tool for explaining the process that was going on between the DHM, the computer, and the music systems, and to demonstrate to onlookers that hand was actually being "read" by the computer. This turned out to be very useful in the long run.

figs/mudra-screen.eps

Figure 37: *PC display for* Bug-Mudra This shows the PC display for debugging and monitoring the DHM performance during *Bug-Mudra.* In the center is a graphics representation of the hand that follows the motion of the DHM. On the left are four sliders representing the four parameters being sent out to the *Hyperlisp* system. On the bottom of the screen appear the bytes as they are streamed out to the *Hyperlisp* system.

Bug-Mudra was premiered in Tokyo in January, 1990 and has since been performed in New York, Los Angeles, Boston, Montreal, Aspen Colorado, and Munich Germany.

11 Recommendations for Future Work

This dissertation defines and documents a new and emerging field of using the hand directly as an input device. Although the specific idea does not originate in this work, nor do many of the techniques, this dissertation is the first to comprehensively present and define whole-hand input as a study in itself, worthy of theoretical and clinical study beyond that of the problems of control systems and human-computer interfaces in general. The immaturity of the field makes the task of narrowing the focus difficult as there is little precedence on which to base careful studies, i.e., there is no theory, and little prior work to spring from.

Many avenues of research became apparent in the course of this dissertation. The experiments performed were chosen as a beginning and as an offering to the mass of unresolved issues in the field. Further exploration is left appropriately for subsequent study. This section suggests several studies that would increase the understanding of the direct use of the hand as an input device.

11.1 Further experimentation and evaluation of hand function

Whole-hand input is not tied to any one application or function. The use of computer-mediated whole-hand input has, for the first time, separated the morphology of hand control from task function. As such, the study of hand capabilities, disjoined from function, is important to the effective design of whole-hand input techniques; particularly as it relates to the intrinsic capabilities of the hand and the statistical variations of abilities across the general population.

Measurement of the precisions of the hand's degrees of freedom

Clinical studies evaluating the precision of human kinesthesia are generally performed in the context of studying sensory mechanisms (Clark and Horch, 1986).[1] It is assumed that all joints have similar mechanisms with varying levels of resolution, and so a few finger joints can be representative of all the hand joints. Although these studies are informative and perhaps data can be extrapolated to the whole hand, they supply a small portion

[1]Clark and Horch (1986) provide a comprehensive review of kinesthetic testing of the human body.

of the information necessary for full evaluation of hand actions for whole-hand input. Comparative data needs to be gathered on all the joints of the hand.

Specific studies hand function in the literature are generally insufficient for the evaluation of hand action. Dexterity studies usually concentrate on the overall ability to perform specific tasks, often related to the workplace, in an effort to evaluate hand disfunction or the effects of clothing or environment on function (Durlach, 1989; Ervin, 1988; Jones, 1989b; Malzahn and Kapur, 1980; Robinette, Ervin, and Zehner, 1986). They do not dissociate the precision of task performance from the precision of the degrees of freedom of the hand.

Joint motion studies usually are limited to range of motion and motion profiles, and do not address the precision of motion (Becker and Thakor, 1988; Chao et al., 1989). Again, this is due to their orientation towards hand impairment and rehabilitation. Joint control studies characteristically use tracking tasks to measure the ability of the subject to control a joint's motion, and quantify the results in terms of the joint's information capacity (Mesplay and Childress, 1988), or report on only a few of the finger joints under specific conditions of motion. Rarely do they address more than one or two joints of the hand.

Hand studies of specific joint function and coordination tend to address the strength or coordination of the joints in a specific task (such as pinching) (Cole, Gracco, and Abbs, 1984; Cole and Abbs, 1986), and not the overall precision of the joint or degree of freedom itself. Joint sensor resolution studies concentrate on the ability of the proprioceptive sensors to detect passive joint motion, i.e. input resolution, not output resolution (Clark et al., 1985), Again, there have been appropriate kinds of studies on other joints of the body, but few evaluating or comparing the joints of the hand (see Clark and Horck, 1986, for details).

The following experiment is proposed to study the issue of resolution and precision of the degrees of freedom of the hand. The results would help in matching hand degrees of freedom to task degrees of freedom on the basis of resolution and precision.[2]

The goal of the experiment is to measure the resolution of the individual degrees of freedom (or joint motions) of the hand. To perform the experiment, a ten-bit scale (0-1023) is put

[2]Experiments reviewed in (Clark and Horch, 1986) also could be adapted to evaluate precision of the degrees of freedom of the hand.

on a high-resolution graphic workstation monitor. Subjects control a slider on the scale by moving the degree of freedom being tested. The joint measuring device must be capable of ten-bit resolution over the range of motion. An accurate goniometer attached to the degree of freedom to be studied should suffice. The goniometer must have a linear response to joint angle and return a ten-bit value that reaches 0 at one joint limit and 1023 at the other. At this point in time the DHM is the only commercial device that is designed to dynamically measure finger joints with this resolution. The ten-bit requirement is based on informal observations of DHM users engaged in similar tasks suggesting that nine bits is an upper limit on control resolution.

figs/joint-res.eps

Figure 38: *Proposed joint resolution experiment* Users are asked to match the slider to a random location on the scale as indicated by the arrow by flexing the degree of freedom being studied.

Subjects are asked to match the slider with a series of random numbers on the scale. After five seconds, the value of the slider is recorded and a new goal is presented. This is repeated often enough to insure statistical significance, say for five minutes (sixty values). The ratio between the error in matching the value and range of the slider indicates the resolution of the degree of freedom. For instance, a mean error of 3 suggests a resolution of approximately 8.4 bits. This is repeated for each of the degrees of freedom of the hand. It should be noted that this differs from a tracking task in that the subject is not following a moving target thus eliminating dynamic effects. The five second time frame is suggested as a compromise between being long enough to avoid the dynamic effects and short enough to prevent fatigue.

Dexterity testing

Based on the discussion above, the ability to coordinate degrees of freedom, and thus dexterity also should be investigated. Again, conventional dexterity testing has evaluated the ability to coordinate degrees only as they apply to the ability to perform a specific function. The goal of the experiments proposed here is to evaluate the ability to coordinate degrees of freedom and to be able to gauge the difficulty of coordinated hand actions.

The experimental setup used in the precision tests can be used for dexterity by measuring the accuracy of more than one degree of freedom. In this experiment multiple sliders appear on the screen, each one controlled by one of the degrees of freedom being tested. Subjects are asked to simultaneously match random values on the sliders. The error in each of the degrees of freedom in coordination compared to the error of the individual degrees of freedom alone (as measured in the precision experiment) gives an indication of the ability to coordinate the specific degrees of freedom. This experiment should be done with all reasonable combinations of degrees of freedom, such as combinations of flexion of MCP joints on different fingers, combinations of abduction and flexion of MCP joints on the same and different fingers, combinations of PIP joints on different fingers, combinations of PIP and MCP joints of the same and different digits, and so on. Although these results could be amalgamated to provide an overall measure of the ability to coordinate degrees of freedom, they are most useful as measures of coordination of specific degree-of-freedom combinations that can guide the design of effective whole-hand input strategies for new tasks.

A problem with this experiment may be that using individual sliders encourages the subjects to decompose the degrees of freedom. This does not invalidate the results, however it would be preferable that the test encouraged the coordination of the degrees of freedom. For example, in the two-degrees of freedom case, matching a random point on an x-y grid would be an improvement over two sliders. A point in space could be used for three degrees of freedom. More degrees of freedom would have to be mapped to another parameter such as the brightness of the point. As the control becomes more complex, the experiment taxes the skills of the subject to perform the task at all, regardless of the interface.

Fidelity and resolution

There is an important difference between fidelity and resolution in whole-hand input devices (see page 95). Fidelity refers to how well the information coming from the sensors reflects the state of the hand. Resolution refers to the number of bits of information available from the device sensors in tracking the hand. Devices can have high resolution but poor fidelity, i.e., the sensors return many bits, but don't track the hand very well; and devices can have low resolution and high fidelity, i.e., few bits, but track the hand very well.

For some tasks, the fidelity of the device may not be important. If the task requires fine control, but the hand motion used is not kinematically relevant, then it may not matter how well the device reflects the true state of the hand. However, if the shape of the hand is helpful or crucial to the performance of the task, then fidelity would be important.

How fidelity and precision trade off, and what effect they have on task performance is not known. Developers of whole hand-input devices need to know where to concentrate money and effort in the building of devices, and designers of whole-hand input techniques need to know what factors are important in the use of whole-hand input for a task. Experiments that tested the importance of fidelity and resolution to particular tasks would provide useful information to both of these processes.

11.2 Evaluating whole hand-input devices

A standard test battery for evaluating the capabilities of whole-hand input devices would provide valuable information to both device engineers and interface designers in building better whole-hand input devices and understanding what are the useful specifications for a whole-hand input device. For example, a standard specification currently used for devices indicates the resolution of the sensors. However, the useful resolution specification is the resolution over the range of finger flexion (and perhaps after sensor noise is filtered out). For the DataGlove, informal evaluation puts sensor and useful resolutions at 8 and 5 bits, for the DHM, 12 and 9, and for the Power Glove, 2 and 2. Using a standard set of metrics (such as the device capabilities in the evaluation guide) to describe whole-hand input devices would help developers discuss, evaluate, and compare hardware choices for whole-hand input use.

11.3 Abstract whole-hand input devices

A distinction can be made between abstract (or virtual) devices and physical devices. The former categorize input behavior with abstractions such as *button*, *pick*, and *valuator*. The latter are physical devices that may behave as one or more abstract devices. Abstract devices describe device-independent interface models. Physical devices can then be adapted to the abstract specification. A mouse, tablet, or trackball can be used interchangeably with an application using an abstract device specification. For example a *signing device* could be an abstract device which allows the user to specify a number of discrete symbols. Depending on the number of symbols required, a signing device could be physically realized with a Power Glove, a DataGlove, a DHM, a bat, a large set of buttons, or a tablet with a large "button template." The only criteria is that a finite number of discrete symbols can be specified with the device. The design method for whole-hand input begins with the assumption of generic whole-hand device input, i.e., an abstract whole-hand input device. Only at the end of the method is a physical device found to match the whole-hand input device specifications.

Current conventions use the GKS set of virtual devices, *pick*, *choice*, *locator*, *valuator*, *stroke*, and *string*. With the emergence of whole-hand input devices, this set needs to be extended to encompass the capabilities of the whole hand. Further experience with whole-hand input devices, and a systematic investigation using the principles of the design method could formalize abstract whole-hand input devices and extend current conventions.

11.4 Computational notations for whole-hand input

One of the difficulties with complex motions such as those of the human body or the hand is developing a notation for those motions. Several systems for movement notation exist in the field of dance, notably Labanotation (Laban, 1975) and Benesh notation, in the study of sign language (Cohen, Namir, and Schlesinger, 1977; Stokoe, 1960), and for hand shape (grips) in robotics (Cutkosky and Wright, 1986). Some attempts have been made to implement computer graphic animation systems using dance notations (Calvert and Chapman, 1978; Singh, 1983), but the notations tend to lack the fine motion specification that is not necessary when directing intelligent humans, but vital when specifying motion to "literal-minded" computers. Nevertheless, the concepts on which the notations are built can serve as basis for the development of movement notations suited for computational use (Badler, 1986).

A notation for whole-hand input could serve as a foundation for the development of the next generation of whole-hand input libraries as well as provide a common basis for the development and communication of whole-hand input techniques. Hand actions could be written in a language that both served as specifications to hand motion interpretation software as well as detailed instructions to a user. With the proper notational conventions, gesture recognition techniques could be developed using the lexical elements of the notation as primitives in the recognition process. Although whole-hand input has a slightly different emphasis from that of sign language and robotics, existing notational schemes may be useful as a basis from which to develop a convention for whole-hand input.

11.5 Incorporating sensory feedback to the design method

This dissertation deals primarily with *input* from hand actions, and does not address the effects of sensory feedback. However, the effects of different forms of feedback can be profound and should be accounted for in the design of interactive systems.

Further work should be directed to extending the design method presented in Section 6 to account for effects of feedback in the human-computer interface. The issues and effects of sensory feedback are discussed in Section 4.5 so they will not be repeated here.

The design method can be extended to incorporate sensory feedback, including visual, auditory, tactile, and kinesthetic feedback, by enlarging the evaluation guide (Section 6.4) with another list for the characteristics of feedback methods. Task characteristics should be expanded to include characteristics relevant to feedback. The individual characteristics of the feedback methods can be iteratively compared with the task characteristics in much the same way as hand action capabilities are compared with task characteristics.

For instance, the characteristics of feedback might be the following:

Sensory feedback

1. Type (tactile, kinesthetic, auditory, visual, etc.)
2. Resolution (bits of perception; acuity)
3. Range (e.g., auditory is 10 Hz. to 22 KHz., 30 dB to 80 dB)
4. Coupling with motor function

   passive vs. active function (e.g., active touch vs. passive touch)

   reaction time (e.g., visual: approx. 250 ms.)

These can be compared with the task characteristics and requirements which have been extended to include the nature of the feedback available from an application. The characteristics of the feedback from an application are a function of the nature of the task (e.g., contact between the robot hand and the environment can or cannot be seen clearly), or feedback mechanisms specifically designed for the task (e.g., touch sensors in the robot gripper transmit forces back to the hand).

Task characteristics and requirements

1. Degrees of freedom

   ⋮

12. Available feedback

    form (tactile, kinesthetic, auditory, visual, etc.)

    resolution (e.g., 10 levels of pressure must be distinguished)

    lag-time[3] (e.g., earth-moon communications)

    range (e.g., sound produced is quiet: 20-40 dB)

[3] Conway, Voltz, and Walker (1990) present a useful collection of methods for dealing with feed-forward and feedback lag times.

Unfortunately, this comparison may not tell the whole story. The synergy of the many parameters of sensory feedback and the sensorimotor actions of the hand is poorly understood. Seemingly minor changes in the form of one or more of the parameters in feedback or hand action may drastically influence the effectiveness of an interaction design. For instance there is an important distinction between active and passive touch in the haptic perception of the environment (Gibson, 1962).

It will be important to continue experimentation with the influence of sensory feedback to task performance. If properly used, sensory feedback can improve the effectiveness of whole-hand input methods. If improperly used, it can have the opposite result. The interaction between visual feedback and haptic input is well represented in the literature, however the role of visual feedback specifically with whole-hand input is yet to be described. Tactile and kinesthetic feedback are the most intriguing of the sensory feedback problems. They are naturally associated with whole-hand input, yet are the least understood, and the most difficult and complex to provide (see discussion in Section 4.5). It is additionally important to study because the tactile and kinesthetic sensorimotor loop is perhaps tighter than any other. Auditory feedback is only partly understood, particularly the extent to which the feedback must be realistic or symbolic (an analogous issue exists for tactile feedback). It is inexpensive to provide, yet may substantially improve an interface.

Extending the design method for sensory feedback will provide a framework in which these issues may be addressed. However, as a design tool, the method will still be dependent on the understanding of the complex interactions in the sensorimotor loop and the resultant entries in the evaluation guide.

11.6 Device improvement

There are many improvements that can be made in whole-hand input device technology. This is not to say that improvements will be technically or economically easy to make, however as more people use whole-hand input the economics will support technological advancements. The following discussion is not meant to be a comprehensive list of device improvements, rather a collection of informal observations based on four years of use of whole-hand input devices and conversations with researchers in the field over that period.

Three-space tracking technology

One of the most bothersome problems with whole-hand input devices is the three-space tracking technologies. The Polhemus tracker is accurate enough for many whole-hand input tasks, but is plagued by the electromagnetic noise common in whole-hand input environments and is hindered by long lag times (Liang, Shaw, and Green, 1991). The Ascension Bird has improved (but not perfect) resistance to electromagnetic interference over the Polhemus and slightly shorter lag times,[4] but still falls short of an ideal lag-free, noise-free, unobtrusive three-space sensor. Ultrasonic systems such as those on the Power Glove and by Logitech require line of sight from the transmitter to the receiver (reducing at least one rotation to $\pm 90°$) and either suffer from acoustic reflections (the Power Glove) or have reduced workspaces (Logitech). Optical solutions also have their problems as discussed in Section 5.

Solutions to this problem are difficult and have plagued researchers for many years. The goal is to provide noise-free, lag-free, accurate, and unobtrusive, tracking of the position and orientation of the hand at speeds high enough for rapid movement (at or greater than 100 Hz.). The system needs to have a range equal to that of the human reach envelope (approximately three cubic meters). In addition, several of the systems should be able to be used in the same physical space without interference with each other (a feature supported by both Ascension and Polhemus at a higher system price).

All existing tracking systems and whole-hand input devices require a cable attachment to the host computer. In most cases the cable is a slight nuisance as it restricts the range of motion of the hand and placement of electronic equipment, as well as getting in the way of gesturing. Future systems would benefit from not having to be tied to large pieces of equipment. For whole-hand input devices that need to be worn on the body, a small transmitter on the hand or belt would free up the user a great deal, as well as allow equipment to be placed at the convenience of the work area.

[4]Informal observations.

Sensor precision, linearity, and calibration

Attaching goniometers to the hand in such a way that they are both accurate and comfortable over wide ranges and speeds of motion is a difficult task. The DataGlove is comfortable, but not very accurate. The DHM is accurate, but not very comfortable. Kramer's glove falls in between the two, while the Power Glove has neither quality. Each one fills a niche in the requirements for a whole-hand input device, but none satisfy all needs. As suggested above in this section, studies of hand precision and the importance of sensor accuracy should guide the development of future whole-hand input devices. On the supposition that the accuracy of sensors relates to hardware cost, future devices should support a range of sensor accuracies from two to ten bits. Those applications that require a few hand signs can use two-bit sensors, those that require sensitive, coordinated control can use ten-bit sensors.

Linearity and independence of sensors are also important factors for accurate tracking of the hand and precision control. As discussed in Section 9, the DataGlove sensors are non-linear and interdependent. The DHM sensors are more linear and independent, but require careful calibration in manufacture and need to be recalibrated periodically.[5] Future devices should pay close attention to sensor linearity in the normal range of use and insure that correlation between sensors is avoided. Sensors should be reliably free from drift over long periods of time (years) or easy to calibrate.

Degrees of freedom

The number of degrees of freedom of general purpose whole-hand input devices should be modular and variable. Different applications require different degrees of freedom. Each degree of freedom has some economic cost and some computational cost. Applications that require few degrees of freedom should be able to take advantage of savings incurred by reducing the degrees of freedom on the devices they use. With a device like the DHM, the number of degrees of freedom relates to the complexity of the hardware, the comfort of the fit on the hand, and the time it takes to don and doff the device. A whole-hand input device with modular degrees of freedom would reduce these problems to an appropriate

[5] Personal experience with one DHM indicates that for every three or four months of use, the sensors should be recalibrated. The process is not difficult and takes three to five minutes per sensor. By contrast, the DataGlove sensors are very difficult to recalibrate.

level for each application.

Force-feedback

As discussed in Section 4.5 there is evidence that force-feedback can improve control and task performance in certain situations (Brooks et al., 1990; Minsky et al., 1990). As new whole-hand input devices are designed, the role of force-feedback should be considered carefully, and provisions made for its inclusion in some tasks.

11.7 Role of external hand constraints

As touched on in Section 4.4, external constraints can aid the use of whole-hand input. The nature and extent of useful external constraints is complex, and has not been studied for whole-hand input. The design method presented in Section 6 will indicate where constraints would be useful for a particular hand action and task primitive, assuming a particular behavior and effectiveness for the constraint. The accuracy of those assumptions is up to the designer. Experiments evaluating task performance with and without constraints such as hand supports and motion damping, would allow designers to make informed decisions in the use of constraints with whole-hand input.

11.8 Hand to work-space mappings

A ripe area for investigation involves the effect of different hand to work-space mappings. Most tasks using direct control (page 81) do not involve one-to-one mappings from hand to task. Thus, linear or non-linear mappings must be used for control in these tasks. The ability of humans to adapt to and manage different mappings, as well as what aids or hinders adaptation is not well understood.

As an example, the experiment involving the control of the six-legged walker's grasping (page 113) subjects did not actually give subjects direct control of the object in the walker's grasp, but set goal orientations that inverse kinematic routines used to move the arm joints. The better the subject understood the kinematic relation between the arms and the orientation of the object, the better the subject performed the task. In unreported preliminary tests, subjects controlled the arms at both high and low level controls. Anecdotal evidence indicates that subjects experienced with whole-hand input and/or robotic kinematics were better able to adapt to the mappings between hand and robot arm than other subjects (see discussion on page 172). The degree of damping (digital low-pass and IIR filtering) of the hand input also affected performance of the task. Subjects reported that too little or too much damping increased task difficulty. Further experimentation with hand to task mappings would provide important contributions to the problems of direct robotic control.

The effect of linear scaling of hand actions is not well understood either. Experiments that varied the gain of hand motion to task motion may reveal optimal gains or limits to the effectiveness of linear mappings. Rotational scaling is most interesting. It would be useful to know how well people can control rotations through linear mappings. For instance, the extent to which control is affected by 90° rotations of the hand causing 360° rotations in the control space.

11.9 Training for whole-hand input

It is not unreasonable to expect effective whole-hand input for some tasks to require operator skill. Personal observation indicates that a majority of people have limited general hand and finger dexterity but with training they can improve their hand function. This is certainly true for specific skill-based, dexterity-demanding endeavors such as playing instruments (especially keyboards), dance, surgery, typing and so on. Many practitioners of these arts regularly perform hand and finger exercises and stretches. Workers using whole-hand input would justifiably benefit from similar regimes. Training programs and exercises from the fields of music (Irwin and Irwin, 1988), and dance can be used as basis for the development of training programs to increase the ability and performance of whole-hand input users.

12 Conclusion

The dissertation shows that the hand can be used as a sophisticated computer input and control device in a wide variety of application domains, providing real-time dexterous control of complex tasks with the coordination of many degrees of freedom. Whole-hand input is most effective when taking advantage of the innate qualities of the hand: naturalness, adaptability, and dexterity. Applications using this kind of input in fields such as robotics, remote control, teleoperation, music, puppetry, and computer animation are illustrated by prototype systems in these domains.

Whole-hand input is not appropriate for all applications, and benefits from careful, analytic and empirical design processes to achieve maximal effectiveness. The *design method for whole-hand input* presented in this dissertation provides a disciplined approach to the design process of providing whole-hand input solutions to control of application tasks. The principles behind the design method have been demonstrated and validated by a series of experiments comparing whole-hand input to conventional input (dials and buttons).

For the practical implementation of whole-hand input it is useful to have an abstract whole-hand input device type and function library. This allows software applications to use a variety of whole-hand input devices with changes to only the low-level device drivers. The use of the library and of whole-hand input in general is demonstrated in this dissertation by four application prototypes.

Whole-hand input is a field of study in itself, apart from (but not without) specific applications or devices. The study of the capabilities of the hand, independent of task, has not had application until the recent emergence of whole-hand input. More must be discovered about the action of the hand apart from task function and the use of whole-hand input in context, to be able to intelligently design future computer-based controls. It is hoped that this dissertation lays some of the groundwork for the successful pursuance of this goal.

Glossary

The definitions in this glossary refer to terms as they are used in this dissertation. Some of the terms may have different meanings in other disciplines.

abduction — Spreading of the fingers away from the middle finger. Opposite of *adduction*. (See Figure 2 on page 25.)

abstract input device — A conceptual or computational description of a method of computer input independent of physical device. The use of abstract input devices allows interactive computer software to be developed independently of physical input devices. This allows different physical devices to be used interchangeably with minimal software modification. Foley and Wallace (1974) discuss this concept in detail, defining four abstract input devices *pick*, *button*, *locator*, and *valuator*. (Also called *logical device* or *virtual device*.)

adduction — Bringing the fingers together towards the middle finger. Opposite of *abduction*. (See Figure 2 on page 25.)

application — A specific project or set of tasks with a common purpose, goal, environment, or equipment. For example, remote operation of a repair robot, or removing a gallbladder. Within applications are individual *tasks* with specific goals.

bolio — The name of a computer program developed by the MIT Media Lab's Computer Graphics and Animation Group for rapid prototyping of interactive simulations using real-time computer graphics. Used in this work for developing prototype applications of whole-hand input. (See page 139.)

cognitive load — The concentration necessary to perform a task to the exclusion of other tasks. Video games have high cognitive loads, driving a car has a medium cognitive load, and walking has a low cognitive load.

constraint — In this work, constraint refers to restrictions and limits to function, motion, or performance. For example, the index finger has the physical constraint that it cannot be extended more than 20° towards the back of the hand, or a *task primitive* may have a time constraint of having to be performed in 45 seconds or less.

control modes — Distinct states of interaction. A control mode embodies a specified set of input procedures and commands. Control modes may be separated along lines of function, levels of abstraction, or methods of use. For example, the six-legged walker described in Section 10.1 has control modes for walking, grasping, high-level control, and low-level control.

DataGlove — A whole-hand input device developed and marketed by VPL Research, Inc. The DataGlove is a lycra glove with lightweight fiber-optic sensors on the fingers and a *Polhemus* three-space tracker on the back of the hand. (See page 65 for a more detailed description.)

degree of freedom — An element of a system that can be varied independently of other elements of the system. The *degrees of freedom* of a system is the minimal number of independently variable elements.

device type — An abstract specification (i.e., not bound to any implementation) of the function and values associated with an *abstract input device*. Analogous to a data type.

dexterity — Skill in using the hands. In this text, *dexterity* also refers specifically to the integration of simple hand motions into higher levels of coordination and competence. For example, manipulating an object in the hand requires dexterity—coordinating simple motions of the individual fingers. If the object is familiar, such as a pencil, the manipulations require little thought, as they have been integrated into a higher level of control (i.e., competence). (See Section 2.2.)

DHM — Dexterous HandMaster. A whole-hand input device originally developed as a *master-slave controller* for the Utah/MIT Dexterous Robot Hand and marketed by Exos, Inc. The DHM is a lightweight framework of sensors that attaches to the hand, recording up to twenty different flexions of the fingers and thumb. (See page 70 for a more detailed description.)

direct manipulation — A style of human-computer interaction characterized by visibility of the objects of interest; rapid, reversible, incremental actions; and replacement of command language syntax by direct manipulation of the object of interest (Shneiderman, 1983).

extension — Straightening a finger joint towards the "open position." Opposite of *flexion*. (See Figure 2 on page 25.)

flexion — Bending a finger joint towards the "closed position." Opposite of *extension*. (See Figure 2 on page 25.)

force-feedback — Forces transmitted to a human operator by a computer to indicate the effect of input (usually by that same person) or the changing state of a process. For example, putting restorative forces on an aircraft control stick if the pilot begins to stall the plane, or "buzzing" the fingertips if the user "touches" an object in a simulated environment. (See *sensory feedback*.)

free-hand motion — Hand motion free from constraint or contact by outside forces, such as from monitoring devices, braces, or tools. All the demonstrations of whole-hand input in this dissertation involve free-hand motion.

hand action — Position, shape, motion, and forces generated by the hand. Used as a general term for descriptions of the hand for whole-hand input. (See Section 6.3.)

haptic — Relating to or based on the sense of touch. In the field of human-computer interaction, *haptic* refers to all effects related to touch, including *kinesthetic* and cutaneous (skin) sensing.

input device — Any physical device used to transfer data, instructions, or signals to a computer process. Examples include keyboard, mouse, data tablet, trackball, joystick, and DataGlove.

kinesthetic — Relating to the sense of joint movement and muscle tensions. The *kinesthetic* sense is mediated by organs located in muscles, tendons, and joints. (In contrast to *tactile*.)

master-slave controller — A device that operates a kinematically similar remote mechanism in a manner such that the motion of the controlling device completely determines the motions of the controlled mechanism. The controlled mechanism takes no action of its own.

"point, reach, and grab" — A form of whole-hand input in which a graphic representation of the user's hand appears on the screen, mimicking the user's motions. The graphic hand can interact with other objects on the screen, allowing the user to manipulate those objects as if they had tangible existence in a physical world. (See Section 4.7.)

Polhemus — Refers to a commonly used three-space tracking device made by Polhemus. The Polhemus uses low-frequency pulsed magnetic fields to sense the six degrees of freedom (three-space position and orientation) of a small sensor relative to a source transmitter. See (Raab et al., 1979) for technical details. A Polhemus is supplied with most DataGloves to track position and orientation of the hand. The small sensor attaches to the back of the hand, while the transmitter is fixed in place nearby (Figure 22). A similar device, called the Bird, is made by Ascension Technologies.

real-time — *n.* The condition in which computer response to external input is continuous and immediate. The term "immediate" is interpreted relative to the application. In this dissertation, 7 Hz. or faster is considered *real-time*, based on the minimal computer graphic update rate that can be used effectively with whole-hand input. *adj.* Occurring, changing, or reacting in *real-time* in response to external input.

sensorimotor — Functioning in or relating to both sensory and motor aspects of bodily activity.

sensory feedback — Purposeful stimulation of the human senses (sight, hearing, touch, smell, taste) by the computer to indicate the effect of the user's input or the changing state of a process. Usually used in conjunction with interactive systems. For example, a computer-generated sound changing pitch as the user flexes the index finger.

supervisory control — Intermittent human control over a computer-mediated process primarily managed by autonomous mechanisms or programs (Sheridan, 1987). *Supervisory control* implies that the human's instructions are at a higher level of control than the autonomous mechanisms. For instance, a mobile robot under supervisory control may be able to navigate to a destination, negotiating terrain and obstacles autonomously, but needs a supervising human to indicate the destination.

tactile — Perceptible by the touch; of or relating to the sense of touch. In the field of human-computer interaction, *tactile* refers to only cutaneous (skin) sensing. (In contrast to *kinesthetic*.)

task — A specific undertaking, job, or chore within an *application*. Tasks accomplish sub-goals of an *application*. For instance, replacing the battery for a radio antenna would be a task within the application of remotely controlling a space repair robot. Tasks can be decomposed into *task primitives*.

task primitive — The smallest executable element of a task (other than individual degrees of freedom) for the purposes of the whole-hand input design method in Section 6. A *task primitive* is usually handled by a single hand action, and analyzed as a unit in the evaluation guide (Section 6.4). For example, *task primitives* in a battery replacement *task* might include unbolting an access hatch, removing the hatch, removing the old battery, storing it away, unpacking the new battery, inserting it into place, and replacing and rebolting the access hatch. (See Section 6.4).

taxonomy — Principles covering the classifying of objects or concepts. Used in the context of this dissertation to refer to an organization and classification of specific concepts relating to whole-hand input. (See Section 6.3.)

virtual input device — See *abstract input device.*

whole-hand input — Information a computer derives from the monitoring of the individual degrees of freedom of the hand. Hand motions for most of this dissertation are considered free of external constraints or contacts. However the definition of whole-hand input includes the use and effects of constraints and contacts. Whole-hand input can also be considered as the direct use of the hand for control of computer-mediated tasks. (See Section 2.)

Glossary

References

Alexander, C., *Notes on the synthesis of form*, Harvard University Press, Cambridge, MA, 1964, Chapter 2.

American Academy of Orthopaedic Surgeons, *Joint motion: Method of measuring and recording*, Churchill Livingstone, New York, 1988.

American Society for Surgery of the Hand, *The hand: Examination and diagnosis*, Churchill Livingstone, New York, 1978.

An, K. N., E. Y. Chao, W. P. Cooney III, and R. L. Linscheid, "Normative model of human hand for biomechanical analysis," *Journal of Biomechanics*, 12, 1979, pp. 775–788.

Badler, Norman I., "Animating human figures: Perspectives and directions," *Proceedings Graphics Interface '86 / Vision Interface '86*, Vancouver, B.C., May 1986, pp. 115–120.

Barnes, J. F., "A task-based metric for telerobotic performance assessment," in *Proceedings of the workshop on space telerobotics (Vol. 2)*, G. Rodriguez (ed.), July 1987, pp. 317–324, (NTIS N89-26492).

Barr, Avron and Edward A. Feigenbaum, eds., *The handbook of artificial intelligence (Vol. 1)*, Addison-Wesley, 1982.

Beaten, Robert J., Richard J. DeHoff, Novia Welman, and Peter H. Hildebrandt, "An evaluation of input devices for 3-d computer display workstations," *Proceedings of SPIE - The International society for optical engineering*, 761, 1987, pp. 94–101.

Becker, Jeff C. and Nitish V. Thakor, "A study of motion of human fingers with application to anthropomorphic designs," *IEEE Transactions on Biomedical Engineering*, 35(2), February 1988, pp. 110–117.

Bejczy, A. K. and J. K. Salisbury, Jr., "Controlling remote manipulators through kinesthetic coupling," *Computers in Mechanical Engineering*, July 1983, pp. 48–60.

Bolt, Richard A., "'Put-that-there': Voice and gesture at the graphics interface," *Computer Graphics*, 14(3), July 1980, pp. 262–270, Proceedings ACM SIGGRAPH '80.

Brooks, Frederick P., Jr., "The computer 'scientist' as toolsmith—studies in interactive computer graphics," in *Information processing 77*, B. Gilchrist (ed.), North Holland Publishing, 1977, pp. 625–634.

Brooks, Frederick P., Jr., Ming Ouh-Young, James J. Batter, and P. Jerome Kilpatrick, "Project GROPE—Haptic displays for scientific visualization," *Computer Graphics*, 24(4), August 1990, pp. 177–185, Proc. ACM SIGGRAPH '90.

Brooks, Martin, "The DataGlove as a man-machine interface for robotics," *The Second IARP Workshop on Medical and Healthcare Robotics*, Newcastle upon Tyne, UK, September 5-7, 1989, pp. 213–225.

Bruderlin, Armin and Thomas W. Calvert, "Goal-directed, dynamic animation of human walking," *Computer Graphics*, 23(3), July 1989, pp. 233–242, Proceedings ACM SIGGRAPH '89.

Burdea, Grigore, Jiachen Zhuang, Ed Roskos, Deborah Silver, and Noshir Langrana, "A portable dexterous master with force feedback," *Presence: Teleoperators and Virtual Environments*, 1(1), January 1992.

Buxton, William, "Lexical and pragmatic considerations of input structures," *Computer Graphics*, 17(1), 1983, pp. 31–37.

Buxton, William, "Communicating with sound (panel presentation)," *Proceedings of CHI'85*, 1985, pp. 115–119.

Buxton, William, "The pragmatics of haptic input," April 1990, ACM CHI'90 Tutorial Notes #26, Seattle, WA.

Buxton, William, Eugene Fiume, Ralph Hill, Allison Lee, and Carson Woo, "Continuous hand-gesture driven input," *Proceedings Graphics Interface '83*, 1983, pp. 191–195.

Buxton, William and Brad A. Myers, "A study of two-handed input," *Proceedings of CHI'86*, 1986, pp. 321–326.

Calvert, T. W. and J. Chapman, "Notation of movement with computer assistance," *Proceedings of the ACM Annual Conference*, 1978, pp. 731–736.

Card, Stuart K., William K. English, and Betty J. Burr, "Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a crt," *Ergonomics*, 21(8), 1979, pp. 601–613.

Card, Stuart K., Jock D. Mackinlay, and George G. Robertson, "The design space of input devices," *Proceedings of CHI'90*, Seattle, WA, April 1990, pp. 117–124.

Carroll, J. M. and R. L. Campbell, *Artifacts as psychological theories: The case of human-computer interaction*, IBM Research Division, T. J. Watson Research Center, RC 13454, #60225, Yorktown Heights, NY, 1988.

Carroll, John M., *Evaluation, description and invention: Paradigms for human-computer interaction*, User Interface Institute, IBM T. J. Watson Research Center, RC 13926, #62583, Yorktown Heights, NY, August 1988.

Chandler, David L., "Robot stand-ins reach out and touch," *The Boston Globe*, Dec. 13, 1990, pp. 35–36.

Chao, Edmund Y. S., Kai-Nan An, William P Cooney III, and Ronald L. Linscheid, *Biomechanics of the hand: A basic research study*, World Scientific Publishing Co., Inc., Teaneck, NJ, 1989.

Chin, K. P. and T. B. Sheridan, "The effect of force feedback on teleoperation," in *Work with computers: Organizational, management, stress and health aspects*, M. J. Smith and G. Salvendy (eds.), Elsevier Science Publishers B.V., Amsterdam, 1989, pp. 505–511, Proceedings of HCI '89, Boston.

Clark, F. J., R. C. Burgess, J. W. Chapin, and W. T. Lipscomb, "Role of intramuscular receptors in the awareness of limb position," *Journal of Neurophysiology*, 54(6), December 1985, pp. 1529–1540.

Clark, Francis J. and Kenneth W. Horch, "Kinesthesia," in *Handbook of perception and human performance, v. 1. Sensory processes and perception*, Kenneth R. Boff, Lloyd Kaufman, and James P. Thomas (eds.), Wiley, New York, 1986, pp. 13.1–13.62.

Cohen, Einya, Lila Namir, and I. M. Schlesinger, *A new dictionary of sign language: employing the Eshkol-Wachmann movement notation system*, Mouton, The Hague, 1977.

Cole, K. J., V. L. Gracco, and J. H. Abbs, "Autogenic and nonautogenic sensorimotor actions in the control of multiarticulate hand movements," *Experimental Brain Research*, 56, 1984, pp. 582–585.

Cole, Kelly J. and James H. Abbs, "Coordination of three-joint digit movements for rapid finger-thumb grasp," *Journal of Neurophysiology*, 55(6), June 1986, pp. 1407–1423.

Conway, Lynn, Richard A. Voltz, and Michael W. Walker, "Teleautonomous systems: Projecting and coordinating intelligent action at a distance," *IEEE Transactions on Robotics and Automation*, 6(2), April 1990, pp. 146–158.

Cosgrove, Tom, "Equipment gets more user friendly," *Engineering news-record*, June 14, 1990, pp. 30–33.

Cutkosky, Mark R. and Paul K. Wright, "Modeling manufacturing grips and correlations with the design of robotic hands," *Proc. IEEE International Conference on Robotics and Automation*, 3, April 1986, pp. 1533–1539.

DeFanti, Thomas A. and Daniel J. Sandin, *Final report to the National Endowment of the Arts*, University of Illinois at Chicago Circle, US NEA R60-34-163, Chicago, IL, 1977.

Delp, David and Scott Delp, "Understanding human movement with computer graphics," *SOMA Engineering for the Human Body*, 3(3), 1989, pp. 17–25.

Delp, S., P. Loan, M. Hoy, F. Zajac, S. Fisher, and J. Rosen, "An interactive graphics-based model of the lower extemity to study orthopaedic surgical procedures," *IEEE Transactions on Biomedical Engineering*, 37(8), August, 1990, Special issue on interaction with and visualization of biomedical data.

Drucker, Steven M., *Evaluation of 2 6-DOF input devices in a 3D graphic environment*, MIT Media Lab, Computer Graphics and Animation Group, Unpublished paper, Cambridge, MA, December 1990.

Durlach, Nathaniel I., *Research on reduced-capability human hands*, M.I.T. Research Laboratory of Electronics, Proposal to Office of Navel Research, Cambridge, MA, 1989.

Eglowstein, Howard, "Reach out and touch your data," *Byte*, July 1990, pp. 283–290.

Eisenberg, Adam, "Ghostbusters revisited," *Cinefex*, 40, November 1989, pp. 4–45.

Enderle, G., K. Kansy, and G. Pfaff, *Computer graphics programming: GKS – the graphics standard*, Springer-Verlag, New York, 1984.

Ervin, Cay A., "A standardized test battery," in *Performance of protective clothing: Second symposium, ASTM STP 1989*, S. Z. Mansdorf, R. Sager, and A. P. Nielsen (eds.), American Society for Testing and Materials, Philadelphia, 1988, pp. 50–56.

Evarts, Edward V., "Sensorimotor cortex activity associated with movements triggered by visual as compared to somesthetic inputs," in *The neurosciences: Third study program*, Massachusetts Institute of Technology, Cambridge, MA, 1974, pp. 327–337.

Feiner, Steven and Clifford Beshers, "Visualizing n-dimensional virtual worlds with n-vision," *Computer Graphics*, 24(2), March 1990, pp. 37–38, Proceedings 1990 Symposium on Interactive Graphics.

Fels, S. Sidney, *Building adaptive interfaces with neural networks: The glove-talk pilot study*, Department of Computer Science, University of Toronto, Technical Report CRG-TR-90-1, Toronto, Canada, February 1990.

Fisher, S. S., M. McGreevy, J. Humphries, and W. Robinett, "Virtual environment display system," *Proc. 1986 ACM Workshop on Interactive Graphics*, Chapel Hill, NC, October 23-24, 1986, pp. 77–87.

Fisher, Scott S., "Virtual environments, personal simulation & telepresence," in *Implementing and interacting with realtime microworlds*, ACM SIGGRAPH '89 Course Notes #29, July 31, 1989.

Fitts, P. M., "The information capacity of the human motor system in controlling amplitude of movement," *Journal of Experimental Psychology*, 47(6), 1954, pp. 381–391.

Flach, John M., "The ecology of human-machine systems I: Introduction," *Ecological Psychology*, 2(3), 1990, pp. 191–205.

Foley, James D. and Victor L. Wallace, "The art of natural graphic man-machine conversation," *Proceedings of the IEEE*, 62(4), April 1974, pp. 462–471.

Furnas, George W., "Generalized fisheye views," *Proceedings CHI'86*, April 1986, pp. 16–23.

Gaver, W., "Auditory icons: Using sound in computer interfaces," *Human Computer Interactions*, 2, 1986, pp. 167–177.

Gialanze, Mark, *The Musicglove*, Music and Cognition Group, Media Lab, Massachusetts Institute of Technology, Unpublished research report, Cambridge, MA, August 1989.

Gibson, James J., "Observations on active touch," *Psychological Review*, 69, 1962, pp. 477–490.

Ginsberg, Carol M. and Delle Maxwell, "Graphical marionette," *Proc. ACM SIGGRAPH/SIGART Workshop on Motion*, Toronto, Canada, April 1983, pp. 172–179.

Girard, Michael, "Interactive design of 3d computer-animated legged animal motion," *IEEE Computer Graphics and Applications*, 7(6), June 1987, pp. 39–51.

Goldstein, E. Bruce, *Sensation and perception (3rd ed.)*, Wadsworth Publishing, Belmont, CA, 1989.

Gómez, Julian E., "Twixt: A 3-d animation system," *Computers & Graphics*, 9(3), 1985, pp. 291–298.

Grimes, Gary J., *Digital data entry glove interface device*, Bell Telephone Laboratories, United States Patent 4,414,537, Murray Hill, NJ, November 8, 1983.

Grissom, S. B., W. E. Carlson, and G. Perlman, "Tablet-based valuators: A comparison of virtual buttons and hand gestures for interactive computer graphics," in *Work with computers: Organizational, management, stress and health aspects*, M. J. Smith and G. Salvendy (eds.), Elsevier Science Publishers B.V., Amsterdam, 1989, pp. 477–483, Proceedings of HCI '89, Boston.

Guidera, Michael G., *The functional significance of the juncturae tendinae in dynamic stabilization of the metacarpophalangeal joints of the hand*, Department of Biomedical Engineering, California State University, Unpublished masters thesis, Sacramento, CA, December 1981.

Hall, Jennifer A., *The human interface in three dimensional computer art space*, Media Lab, Massachusetts Institute of Technology, Unpublished MSVS Thesis, Cambridge, MA, October 1985.

Hanrahan, Pat and David Sturman, "Interactive control of parametric models," *The Visual Computer*, 1(4), December 1985, pp. 260–266.

Heinlein, Robert A. (writing as Anson McDonald), "Waldo," *Astounding Science-Fiction*, 29(6), August 1942, pp. 9–53, Currently available in Waldo and Magic, Inc. by Robert A. Heinlein, Del Ray (Ballantine), 1986.

Henry, Tyson R., Scott E. Hudson, Andrey K. Yeatts, Brad A. Nyers, and Steven Feiner, "A nose gesture interface device: Extending virtual realities," *Proc. UIST '91: ACM Symposium on User Interface Software and Technology*, Hilton Head, South Carolina, Nov 11-13, 1991, pp. 65–68.

Hong, Jiawei and Xiaonan Tan, "Calibrating a VPL DataGlove for teleoperating the Utah/MIT Hand," *Proc. IEEE International Conference on Robotics and Automation*, 3, 1989, pp. 1752–1757.

Hughes, P. J., D. G. Alciatore, J. T. O'Connor, and A. E. Traver, "Construction manipulator operation with "ergosticks"," in *Work with computers: Organizational, management, stress and health aspects*, M.J. Smith and G. Salvendy (eds.), Elsevier Science Publishers B.V., Amsterdam, 1989, pp. 571–578, Proceedings of HCI '89, Boston.

Hunter, J. M., L. H. Schneider, E. J. Mackin, and A. D. Callahan, eds., *Rehabilitation of the hand (2nd ed.)*, CV Mosby, St. Louis, 1984.

Hutchins, Edwin L., James D. Hollan, and Donald A. Norman, "Direct manipulation interfaces," in *User centered system design*, D. A. Norman and S. W. Draper (ed.), Lawrence Erlbaum Associates, Inc., Hillside, NJ, 1986, pp. 87–124.

Irwin, Gregory G. and Lorraine C. Irwin, *Finger fitness: The art of finger control*, Banner Press, 1988.

Iwata, Hiroo, "Artificial reality with force-feedback: Development of desktop virtual space with compact master manipulator," *Computer Graphics*, 24(4), August 1990, pp. 165–170, Proceedings ACM SIGGRAPH '90.

Jacobsen, S. C., E. K. Iversen, D. F. Knutti, R. T. Johnson, and K. B. Biggers, "Design of the Utah/MIT Dexterous Hand," *Proc. IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 1986, pp. 1520–1532.

Jacobsen, S. C., F. M. Smith, E. K. Iversen, and D. K. Backman, "High performance, high dexterity, force reflective teleoperator," *American Nuclear Society 1990 Winter Meeting*, Washington, DC, November 1990.

Jones, Dylan, "The sonic interface," in *Work with computers: Organizational, management, stress and health aspects*, M.J. Smith and G. Salvendy (eds.), Elsevier Science Publishers B.V., Amsterdam, 1989a, pp. 383–388, Proceedings of HCI '89, Boston.

Jones, Lynette A., "The assessment of hand function: a critical review of techniques," *Journal of Hand Surgery*, 14A(2, Pt. 1), March 1989b, pp. 221–228.

Kangari, Roozbeh and Daniel W. Halpin, "Potential robotics utilization in construction," *Journal of Construction Engineering and Management*, 115(1), March 1989, pp. 126–143.

Kaplan, Emanuel B., *Functional and surgical anatomy of the hand (2nd ed.)*, J. B. Lippencott Company, Philadelphia, 1965.

Kaufman, Arie and Roni Yagel, "Tools for interaction in three dimensions," in *Work with computers: Organizational, management, stress and health aspects*, M.J. Smith and G. Salvendy (eds.), Elsevier Science Publishers B.V., Amsterdam, 1989, pp. 469–475, Proceedings of HCI '89, Boston.

Keele, Steven W., "Learning and control of coordinated motor patterns: The programming perspective," in *Human motor behavior: An introduction*, J. A. Scott Kelso (ed.), Lawrence Erlbaum Associates, Hillsdale, NJ, 1982, pp. 161–186.

Kelly, Kevin, Adam Heilbrun, and Barbara Stacks, "An interview with Jaron Lanier: Virtual reality," *Whole Earth Review*, Fall 1989, pp. 108–119.

Kelso, J. A. Scott, ed., *Human motor behavior: An introduction*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1982.

Kilpatrick, Paul Jerome, *The use of kinesthetic supplement in an interactive system*, Computer Science Department, Unpublished doctoral dissertation, University of North Carolina at Chapel Hill, 1976.

Kim, J., "On-line gesture recognition by feature analysis," *Proceedings of Vision Interface*, 1988, pp. 51–55.

Kim, Won S., Frank Tendick, Stephen R. Ellis, and Lawrence W. Stark, "A comparison of position and rate control for telemanipulations with consideration manipulator system dynamics," *IEEE Journal of Robotics and Automation*, RA-3(5), October 1987, pp. 426–436.

Kim, Won S., Frank Tendick, and Lawrence W. Stark, "Visual enhancements in pick-and-place tasks: Human operators controlling a simulated cylindrical manipulator," *IEEE Journal of Robotics and Automation*, RA-3(5), October 1987, pp. 418–425.

Klima, E. S. and U. Bellugi, *The signs of language*, Harvard University Press, Cambridge, MA, 1979.

Kohonen, Teuvo, *Self-organization and associative memory*, Springer-Verlag, 1984.

Kramer, James and Larry Leifer, *The Talking Glove: An expressive and receptive "verbal" communication aid for the deaf, deaf-blind, and nonvocal*, Stanford University, Department of Electrical Engineering, 1989.

Krueger, Myron W., *Artificial reality*, Addison-Wesley, Reading, MA, 1983, Out of print..

Krueger, Myron W., *Artificial reality (2nd ed.)*, Addison-Wesley, Reading, MA, 1990.

Laban, Rudolf, *Laban's principles of dance and movement notation (2nd ed.)*, Plays, Inc., Boston, 1975, Published in 1956 and 1970 under the title: *Principles of dance and movement notation.*

Laurel, Brenda K., "Interface as mimesis," in *User centered system design*, D. A. Norman and S. W. Draper (eds.), Lawrence Erlbaum Associates, Inc., Hillside, NJ, 1986, pp. 67–85.

Lederman, Susan J. and Roberta L. Klatzky, "Hand movements: A window into haptic object recognition," *Cognitive Psychology*, 19, 1987, pp. 342–368.

Levit, Creon and Steve Bryson, "A virtual environment for exploration of three-dimensional flowfields," *SPIE/IS&T Symposium on Electronic Imaging Science and Technology: Conf. 1457 – Stereoscopic Displays and Applications II*, San Jose, CA, February 1991.

Liang, Jiandong, Chris Shaw, and Mark Green, "On temporal-spatial realism in the virtual reality environment," *Proc. UIST '91: ACM Symposium on User Interface Software and Technology*, Hilton Head, South Carolina, Nov 11-13, 1991, pp. 19–25.

Machover, Tod, *Flora*, Bridge Records, Inc, 1990.

Mackinlay, Jock D., Steurat K. Card, and George G. Robertson, "Rapid controlled movement through a virtual 3D workspace," *Computer Graphics*, 24(4), August 1990, pp. 171–176, Proceedings ACM SIGGRAPH '90.

Makower, J., M. Parnianpour, and M. Nordin, "The validity assessment of the dexterous hand master: A linkage system for the measurement of joints in the hand," *Abstracts of the First World Congress of Biomechanics (Volume II)*, La Jolla, California, September 1990, p. 338.

Malzahn, Don and Rajiv Kapur, "An ability evaluation system for persons with physical disabilities," *Proceedings of the Human Factos Society—24th Annual Meeting*, 1980, pp. 114–118.

Mann, Robert W. and Erik K. Antonsson, "Gait analysis—precise, rapid, automatic, 3-d position and orientation kinematics and dynamics," *BULLETIN of the Hospital for Joint Diseases Orthopaedic Institute*, XLIII (2), 1983, pp. 137–146.

Marcus, Beth A. and Philip J. Curchill, "Sensing human hand motions for controlling dexterous robots," *The Second Annual Space Operations Automation and Robotics Workshop, held at Wright State University*, July 20-23, 1988, Sponsored by NASA and the USAF.

Martin, Gale, James Pittman, Kent Wittenburg, Richard Cohen, and Tom Parish, "Sign here, please," *Byte*, July 1990, pp. 243–251.

Mason, Matthew. T. and J. Kenneth Salisbury, Jr., *Robot hands and the mechanics of manipulation*, M.I.T. Press, Cambridge, MA, 1985.

McCormick, Bruce H., Thomas A. DeFanti, and Maxine D. Brown, eds., "Visualization in scientific computing," *Computer Graphics*, 21(6), November 1987.

McKenna, Michael and David Zeltzer, "Dynamic simulation of autonomous legged locomotion," *Computer Graphics*, 24(4), August 1990, pp. 29–38, Proceedings ACM SIGGRAPH '90.

McKinnon, G. M., M. L. King, and D. Runnings, "Multi-axis control of telemanipulators," *Proc. First European In-Orbit Operations Technology Symposium*, Darmstaadt, W. Germany, November 1987, pp. 487–491.

Mesplay, K. P. and D. S. Childress, "Capacity of the human operator to move joints as control inputs to prostheses," *Modeling and Control Issues in Biomechanical Systems*, Chicago, IL, November 1988, pp. 17–25, Presented at the Annual Meeting of the American Society of Mechanical Engineers.

Minsky, Margaret, Ming Ouh-young, Oliver Steele, Frederick P. Brooks, Jr., and Max Behensky, "Feeling and seeing: Issues in force display," *Computer Graphics*, 24(2), March 1990, pp. 235–243, Proc. 1990 Symposium on Interactive 3D Graphics, Snowbird, UT.

Minsky, Marvin, "Telepresence," *Omni*, June 1980, pp. 45–50.

Minsky, Marvin, "Nasa held hostage," *Ad Astra*, June 1990, pp. 34–37.

Mishra, Bud and Naomi Silver, "Some discussion of static gripping and its stability," *IEEE Transactions on Systems, Man, and Cybernetics*, 19(4), July/August 1989, pp. 783–796.

Morris, Desmond, *Manwatching*, Harry N. Abrams, New York, 1977.

Napier, John, *Hands*, Pantheon Books, New York, 1980.

Noll, A. Michael, "Man-machine tactile communication," *SID Journal (Society for Information Display)*, 1(2), July/August 1972.

Ouh-young, Ming, *Force display in molecular docking*, Department of Computer Science, Unpublished doctoral dissertation (TR90-004), University of North Carolina at Chapel Hill, February 1990.

Pao, Lucy and Thomas H. Speeter, "Transformation of human hand positions for robotic hand control," *Proc. IEEE International Conference on Robotics and Automation*, 3, 1989, pp. 1758–1763.

Parke, Frederick I., "Parameterized models for facial animation," *IEEE Computer Graphics and Applications*, 2(9), November 1982, pp. 61–68.

Patrick, Nicholas J. M., Tom Sheridan, Mike Massiminio, and Beth A. Marcus, "Design and testing of a non-reactive, fingertip, tactile display for interaction with remote environments," *Proceedings of the SPIE Symposium on Advances in Intelligent Systems*, Boston, MA, November 1990.

Pew, Richard W., "Human perceptual-motor performance," in *Human information processing: Tutorials in performance and cognition*, Barry H. Kantowitz (ed.), Lawrence Erlbaum Associates, Inc., Hillside, NJ, 1974, pp. 1–39.

Pieper, Stephen D., *CAPS: Computer-aided plastic surgery*, Media Lab, Massachusetts Institute of Technology, PhD Thesis, Cambridge, MA, February 1992.

Poizner, H., E. S. Klima, U. Bellugi, and R. B. Livingston, "Motion analysis of grammatical processes in visual-gestural language," *Proc. ACM SIGGRAPH/SIGART Workshop on Motion*, Toronto, Canada, April 1983, pp. 148–171.

Purcell, Patrick, "Gestural input to interactive systems," *Computer Bulletin*, September 1985, pp. 3–7.

Raab, Frederick H., Ernest B. Blood, Terry O. Steiner, and Herbert R. Jones, "Magnetic position and orientation tracking system," *IEEE Transactions on Aerospace and Electronic Systems*, AES-15(5), September 1979, pp. 709–718.

Reynolds, Craig W., "Computer animation with scripts and actors," *Computer Graphics*, 16(3), July 1982, pp. 289–296, Proceedings ACM SIGGRAPH '81.

Robertson, Barbara, "Mike, the talking head," *Computer Graphics World*, July 1988, pp. 15–17.

Robinette, K. M., C. Ervin, and G. F. Zehner, *Dexterity testing of chemical defense gloves(U)*, Armstrong Aerospace Medical Research Lab, AAMRL-TR-86-021, Wright-Patterson AFB, OH, May 1986, (NTIS AD-A173545).

Robinette, Kathleen M., Cay Ervin, and Gregory Zehner, *Development of a standard dexterity test battery*, Armstrong Aerospace Medical Research Lab, AAMRL-TR-87-034, Wright-Patterson AFB, OH, April 1987, (NTIS AD-A188314).

Robotics World, "Teleoperated manipulators aid underwater vehicle operation," 7(1), Jan-Feb 1989, pp. 21–22.

Rogowitz, B. E., "The human visual system: A guide for the display technologist," *Proceedings of the SID*, 24(3), 1983, pp. 235–252.

Rokeby, David, "Very nervous system," Atlanta, GA, August 1988, Interactive Insallation, SIGGRAPH'88 Art Show.

Rolfe, J. M. and K. J. Staples, eds., *Flight simulation*, Cambridge University Press, Cambridge, UK, 1986.

Rubine, Dean, "Specifying gestures by example," *Computer Graphics*, 25(4), July 1991, pp. 329–337, Proceedings ACM SIGGRAPH '91.

Salisbury, K., "Issues in human/computer control of dexterous remote hands," *Proceedings on the Workshop on Space Telerobotics (Vol. II)*, July 1, 1987, pp. 351–359, (NTIS N89-26492).

Sexton, George A., "Cockpit-crew systems design and integration," in *Human Factors in Aviation*, Earl L. Wiener and David C. Nagel (eds.), Academic Press, New York, 1988, pp. 495–526.

Shapiro, Lawrence K. and Howard I. Shapiro, "Construction cranes," *Scientific American*, 258(3), March 1988, pp. 72–79.

Sheridan, Thomas B., "Supervisory control," in *Handbook of Human Factors*, Gavriel Salvendy (ed.), Wiley, New York, 1987, pp. 1243–1268.

Sheridan, Thomas B., "Merging mind and machine," *Technology Review*, October 1989, pp. 33–40.

Shneiderman, Ben, "The future of interactive systems and the emergence of direct manipulation," *Behavior and Information Technology*, 1, 1982, pp. 237–256.

Shneiderman, Ben, "Direct manipulation: A step beyond programming languages," *IEEE Computer*, 16(8), August 1983, pp. 57–69.

Singh, A., *A computerized editor for Benesh movement notation*, University of Waterloo, Unpublished master's thesis, Waterloo, Ontario, 1983.

Skibniewski, Miroslaw J. and Jeffrey S. Russell, "Robotic applications to construction," *Cost Engineering*, 31(6), June 1988, pp. 10–18.

Speeter, Thomas H., *Transforming human hand motion for telemanipulation*, AT&T Bell Laboratories, Technical Memorandum, Holmdel, NJ, Sept. 19, 1989, (Submitted to IEEE-SMC 10/89).

Stokoe, William C., Jr., *Sign language structure: An outline of the visual communication systems of the American deaf*, Studies in Linguistics: Occasional Papers, 8, University of Buffalo, Department of Anthropology and Linguistics, Buffalo, NY, 1960, (Reissued Washington, D.C., Gallaudet College Press).

Stone, Robert J., "The UK Virtual Reality & Telepresence Project: one year on," *Proceedings of the Blenheim Online Conference*, London, November 1991, pp. 131–140.

Sturman, David, David Zeltzer, and Steven Pieper, "Hands-on interaction with virtual environments," *Proc. UIST '89: ACM SIGGRAPH/SIGCHI Symposium on User Interface Software and Technology*, Williamsburg, VA, Nov 13-15, 1989, pp. 19–24.

Sutton, R., J. E. Cherrington, and D. R. Towill, "Manual control of crane systems," *Ergonomics*, 29(10), 1986, pp. 1155–1171.

Takahashi, Tomoichi and Fumio Kishino, *Hand gesture coding based on experiments using a hand gesture interface device*, ATR Communication Systems Research Laboratories, Technical Report, Kyoto, Japan, 1990.

Takemura, H., A. Tomono, and Y. Kobayashi, "An evaluation of 3-d object pointing using a fields sequential steroscopic display," *Proceedings Graphics Interface '88*, Edmonton, 1988.

Trubitt, David, "Into new worlds; virtual reality and the electronic musician," *Electronic Musician*, July 1990, pp. 31–40.

Tubiana, Raoul, "Architecture and functions of the hand," in *The hand*, Raoul Tubiana (ed.), Saunders, Philadelphia, PA, 1981, pp. 19–93.

Tuller, Betty, M. T. Turvey, and Hollis L. Fitch, "The Bernstein perspective: II. The concept of muscle linkage or coordinative structure," in *Human motor behavior: An introduction*, J. A. Scott Kelso (ed.), Lawrence Erlbaum Associates, Hillsdale, NJ, 1982, pp. 253–270.

Turvey, M. T., Hollis L. Fitch, and Betty Tuller, “The Bernstein perspective: I. The problems of degrees of freedom and context-conditioned variability,” in *Human motor behavior: An introduction*, J. A. Scott Kelso (ed.), Lawrence Erlbaum Associates, Hillsdale, NJ, 1982, pp. 239–252.

VPL Research, Inc., *DataGlove model 2 users manual*, Redwood City, CA, 1987.

Vicente, Kim J. and Jens Rasmussen, “The ecology of human-machine systems II: Mediating “direct perception” in complex work domains,” *Ecological Psychology*, 2(3), 1990, pp. 207–249.

Viviani, P. and C. Terzuolo, “Space-time invariance in learned motor skills,” in *Tutorials in motor behavior*, George E. Stelmach and Jean Requin (eds.), Elsevier Science Publishing Co., Inc., New York, 1980, pp. 525–533.

Vivid Effects, Inc., *The Mandala system*, Toronto, Canada, 1989.

W Industries, W Industries, Press release, Leicester, UK, 1991.

Waldron, Manjula B. and Darius Simon, “Parsing method for signed communication,” *IEEE Engineering in Medicine and Biology Society 11th Annual International Conference*, 1989, pp. 1798–1799.

Walters, Graham, “The story of Waldo C. Graphic,” in *3d character animation by computer*, ACM SIGGRAPH ’89 Course Notes #4, July 1989, pp. 65–79.

Ware, Colin, “Using the hand position for virtual object placement,” *The Visual Computer*, 6, 1990, pp. 245–253.

Ware, Colin and Danny R. Jessome, “Using the Bat: A six-dimensional mouse for object placement,” *IEEE Computer Graphics and Applications*, 8(6), 1988, pp. 65–70.

Ware, Colin and Steven Osborne, “Exploration and virtual camera control in virtual three dimensional environments,” *Computer Graphics*, 24(2), March 1990, pp. 175–183, Proceedings 1990 Symposium on Interactive Graphics.

Weimer, David and S. K. Ganapathy, “A synthetic visual environment with hand gesturing and voice input,” *Proceedings CHI’89*, May 1989, pp. 235–240.

Whiteside, J. and D. Wixon, “Improving human-computer interaction—a quest for cognitive science,” in *Interfacing thought: Cognitive aspects of human-computer interaction*, J. M. Carroll (ed.), Breadford/M.I.T. Press, Cambridge, MA, 1987, pp. 337–352.

Williams, Lance, “Bbop,” in *Seminar on three-dimensional computer animation*, ACM SIGGRAPH ’82 Course Notes, July 27, 1982.

Winograd, T. and F. Flores, *Understanding computers and cognition: A new foundation for design*, Ablex, Norwood, NJ, 1986.

Wise, Sam, William Gardner, Eric Sabelman, Erik Valainis, Yuriko Wong, Karen Glass, John Drace, and Joseph Rosen, “Evaluation of a fiber optic glove for semi-automated goniometric measurements,” *Journal of Rehabilitation Research and Development*, 27(4), 1990, pp. 411–424.

Wixon, Dennis and Michael Good, “Interface style and eclecticism: Moving beyond categorical approaches,” *Proceedings of the Human Factors Society—31st Annual Meeting*, 1987, pp. 571–575.

Wolf, Catherine G. and Palmer Morrel-Samuels, “The use of hand-drawn gestures for text editing,” *International Journal of Man-Machine Studies*, 27, 1987, pp. 91–102.

Young, Douglas E. and Richard A. Schmidt, “Motor programs as units of movement control,” in *Making them move: Mechanics, control and animation of articulated figures*, Norman I. Badler, Brian A. Barsky, and David Zeltzer (eds.), Morgan Kaufmann Publishers, San Mateo, CA, 1991, pp. 129–155.

Zeltzer, David, “Towards an integrated view of 3-d computer animation,” *The Visual Computer*, 1(4), December 1985, pp. 249–259.

Zeltzer, David, Steven Pieper, and David Sturman, “An integrated graphical simulation platform,” *Proceedings Graphics Interface ’89*, London, Ontario, June 1989, pp. 266–274.

Zimmerman, Thomas G., Jaron Lanier, Chuck Blanchard, Steve Bryson, and Young Harvill, “A hand gesture interface device,” *Proc. Human Factors in Computing Systems and Graphics Interface (CHI+GI’87)*, Toronto, Canada, April 1987, pp. 189–192.

Zimmerman, Thomas G. and Jaron Z. Lanier, *Computer Data Entry and Manipulation Apparatus and Method*, VPL Research, Inc., United States Patent 4,988,981, Redwood City, CA, January 29, 1991.