A Survey of Hand Posture and Gesture Recognition Techniques and Technology

Joseph J. LaViola Jr.

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-99-11
June 1999

A Survey of Hand Posture and Gesture Recognition Techniques and Technology

Joseph J. LaViola Jr.
Brown University
NSF Science and Technology Center for
Computer Graphics and Scientific Visualization
Box 1910, Providence, RI 02912 USA
jjl@cs.brown.edu

Abstract

This paper surveys the use of hand postures and gestures as a mechanism for interaction with computers, describing both the various techniques for performing accurate recognition and the technological aspects inherent to posture- and gesture-based interaction. First, the technological requirements and limitations for using hand postures and gestures are described by discussing both glove-based and vision-based recognition systems along with advantages and disadvantages of each. Second, the various types of techniques used in recognizing hand postures and gestures are compared and contrasted. Third, the applications that have used hand posture and gesture interfaces are examined. The survey concludes with a summary and a discussion of future research directions.

Contents

1 Introduction

The purpose of this document is to provide a broad introduction to the field of hand posture and gesture recognition as a mechanism for interaction with computers. It presents an extensive survey of all the problems and issues relevant to using hand postures and gestures in user interfaces, consolidating existing information in the field and organizing it in a clear and efficient manner. It also gives a critical review of the information presented so as to point out the general advantages and disadvantages of the various recognition techniques and systems.

Although other surveys have been written on various subsets of hand posture and gesture recognition[99][112], this one is more comprehensive and up-to-date. It is intended to be a starting point for anyone interested in using hand postures and gestures in their interfaces and to give researchers a starting point for exploring the many open research issues.

Although hand postures and gestures are often considered identical, there are distinctions between them. A hand posture is defined as a static movement. For example, making a fist and holding it in a certain position is considered a posture. With a simple posture, each of the fingers is either extended or flexed but not in between; for example a fist, pointing, and thumb's up. With a complex posture, the fingers can be bent at angles other than zero or ninety degrees. Complex postures include various forms of pinching, the "okay" sign and many of the postures used in finger spelling[77].

A gesture is defined as a dynamic movement, such as waving good-bye. Simple gestures are made in two ways. The first way involves a simple or complex posture and change in the position or orientation of the hand, such as making a pinching posture and changing the hand's position. The second way entails moving the fingers in some way with no change in the position and orientation of the hand, for example, moving the index and middle finger back and forth to urge someone to move closer. A complex gesture is one that includes finger movement, wrist movement and changes in the hand's position and orientation. Many of the signs in American Sign Language are examples of this type of gesture.

An important criterion in discussing this interaction paradigm is the number of postures and gestures that a given recognition system or algorithmic technique can accurately recognize[1]. In this paper, 1 to 15 postures and gestures is considered a small set, 15 to 25 is medium-sized, and anything over 25 is considered large.

The remainder of this survey is divided into five main parts. The first part, Section 2, talks about the various aspects of hand posture and gesture recognition technology. It discusses a number of current glove-based input devices and the advantages and disadvantages of each. It also describes aspects of vision-based recognition systems and compares and contrasts both vision- and glove-based systems. Section 3 describes the various algorithms used in hand posture and gesture recognition and discusses the advantages and disadvantages of each. Section 4 describes the many applications that have used hand postures and gestures in their interfaces. Section 5 presents conclusions and areas of future research. As an addendum, two appendices are also provided which describe the basic anatomical structure of the hand and the various classification systems and taxonomies used in describing hand postures and gestures. The purpose of these appendices is to provide the reader with some terminology that frequently occurs in the literature.

[1]Unfortunately, accuracy is a relative measure. One definition of how accurate a recognition system is can be entirely different from another. For the purposes of this paper, techniques and algorithms with accuracy measures over 90% can generally be considered accurate. However, whether such a system is usable is subject to debate.

2 Hand Posture and Gesture Recognition Technology

This section discusses the requirements for hand posture and gesture recognition. It describes the two main solutions for collecting the required data to perform recognition, the glove-based solution and the camera- or vision-based solution, and looks at the advantages and disadvantages of each.

2.1 Data Collection for Hand Postures and Gestures

The first step in using hand posture and gestures in computer applications is gathering raw data. This raw data is then analyzed by using various recognition algorithms (see Section 3) to extract meaning or context from the data in order to perform tasks in the application. Raw data is collected in three ways. The first is to use input devices worn by the user. This setup usually consists of one or two instrumented gloves that measure the various joint angles of the hand and a six degree of freedom (6 DOF) tracking device that gathers hand position and orientation data. The second way to collect raw hand data is to use a computer-vision-based approach by which one or more cameras collect images of the user's hands. The cameras grab an arbitrary number of images per second and send them to image processing routines to perform posture and gesture recognition as well as 3D triangulation to find the hands' position in space. The third way to collect raw hand data is to combine the previous two methods in a hybrid approach with the hope of achieving a more accurate level of recognition by using the two data streams to reduce each other's error. Very little work has been done on hybrid tracking for hand posture and gesture recognition, but this type of tracking has been successful in augmented reality systems like Auer[7] and State[98], and could well be applied to hand posture and gesture recognition.

2.2 Data Collection Using Instrumented Gloves and Trackers

Raw data collection using instrumented gloves and trackers requires users to physically attach computer input devices to their hands. The instrumented gloves report data values for the movement of the fingers; the amount of reported data values depends on the type of glove worn. The trackers are attached to the back of the hand or the upper wrist, depending on the type of glove worn, and give back data on the position and orientation of the hand in 3D space.

2.2.1 Tracking Devices

A number of different tracking technologies are available to track hand position and orientation. This survey touches on the most common; for a detailed discussion see Youngblut's review of virtual environment interface technology[120], Encarnação's survey on input technology[31] or Mulder's survey on human movement technology[75]. These three papers present a very thorough analysis of over 25 different tracking devices on the market today.

Three non-vision-based methods for tracking hand position and orientation are magnetic, acoustic, and inertial tracking. With magnetic tracking, a transmitting device emits a low-frequency magnetic field from which a small sensor, the receiver, determines its position and orientation relative to a magnetic source. The advantages of these types of systems are that they have good range, anywhere from fifteen to thirty feet away if some extended range hardware is used, are generally accurate to within 0.1 inches in position and 0.1 degrees in orientation, and are moderately priced [6][86]. Their main disadvantage is that any ferromagnetic or conductive objects present in the room with the transmitter will distort the magnetic field reducing the accuracy. The distortion can be handled with filtering algorithms, but doing so introduces a more complex computational component and increases latency. The two most commonly used magnetic trackers today are from Polhemus and Ascension Technology Corporation.

Acoustic tracking systems or ultrasonic tracking uses high-frequency sound emitted from a source component that is placed on the hand or area to be tracked. Microphones placed in the environment receive ultrasonic pings from the source components to determine their location and orientation[99]. In most cases, the microphones are placed in a triangular array and this region determines the area of tracked space. The advantages of acoustic tracking systems are that they are relatively inexpensive and lightweight. However, these devices have a short range and their accuracy suffers if acoustically reflective surfaces are present in the room. Another disadvantage of acoustic tracking is that external noises such as jingling keys or a ringing phone can cause interference in the tracking signal and thus reduce accuracy. Logitech's acoustic tracking systems seem to be the most commonly used; however, some newer companies like Freepoint 3D have entered this field[31]. Acoustic tracking has also been incorporated into some glove-based devices such as the Mattel Power Glove™ [99] and VPL's Z-Glove™ [121], discussed in further detail in Section 2.2.2.

Finally, inertial tracking systems use a variety of inertial measurement devices such as gyroscopes, servo-accelerometers, and even micromachined quartz tuning forks that sense angular velocity using the Coriolis principle[31]. The advantages of an inertial tracking system is speed, accuracy and range, but the major problems with these systems are that they usually only track three degrees of freedom (either position or orientation data) and they suffer from gyroscopic drift. The most commonly used inertial tracking systems are InterSense's IS-300 and IS-600. The IS-300 measures only orientation data but uses gravitometer and compass measurements to prevent accumulation of gyroscopic drift and employs a motion prediction mechanism that predicts motion up to 50 milliseconds in the future. The IS-600 tracks both position and orientation but requires an additional ultrasonic component to acquire the position data[46].

A common problem with these tracking devices is that they do not have perfect accuracy. A promising way of achieving greater accuracy is to use prediction/correction techniques to filter the position and orientation data. One of the most widely used filtering techniques is the Kalman filter, a recursive mathematical procedure that uses the predictor/corrector mechanism for least-squares estimation for linear systems. Welch and Bishop[114] and Maybeck[70] both provide detailed discussions and mathematical derivations of the Kalman filter for the interested reader. Kalman filtering can be applied to tracking devices, vision tracking[10], and hybrid tracking systems as well[113].

2.2.2 Instrumented Gloves

Instrumented gloves measure finger movement through various kinds of sensor technology[1]. These sensors are embedded in a glove or placed on it, usually on the back of the hand. Glove-based input devices can be broadly divided into those gloves that are available in the marketplace today and those that are not, either because their respective companies have gone out of business or because they were never developed commercially. Both Sturman[101] and Kadous[48] discuss both categories of gloves, but their surveys are know out of date. Encarnação[31] and Youngblut's[120] discussions of glove input devices deal specifically with those currently available from commercial vendors. The present survey gives both a historical perspective on these devices by describing those gloves that are no longer available and a practical guide to those gloves that are on the market today.

Historical Perspective One of the first instrumented gloves described in the literature was the 'Sayre Glove' developed by Thomas Defanti and Daniel Sandin in a 1977 project for the National Endowment of the Arts[26]. This glove used light-based sensors with flexible tubes with a light source at one end and a photocell at the other. As the fingers were bent, the amount of light that hit the photocells varied thus providing a measure of finger flexion. The glove could measure the metacarpophalangeal joints of the four fingers and thumb along with the proximal interphalangeal joints of the index and middle fingers, for a total of 7 DOF (Figure A.1 shows a diagram of the joints of

[1] The exception to this definition is the Fakespace Pinch™ Glove. Instead of measuring finger movement, Pinch gloves detect electrical contact made when the fingertips touch.

the hand). It was designed for multidimensional control of sliders and other 2D widgets and did not have the sophistication or accuracy needed for hand posture or gesture recognition.

The Digital Data Entry Glove, designed by Gary Grimes at Bell Telephone Laboratories in 1981, was invented specifically for performing manual data entry using the Single-Hand Manual Alphabet[41]. It used touch or proximity sensors, “knuckle-bend sensors”, tilt sensors, and inertial sensors to replace a traditional keyboard. The touch or proximity sensors determined whether the user’s thumb was touching another part of the hand or fingers. They were made of silver-filled conductive rubber pads that sent an electrical signal when they made contact. The four knuckle-bend sensors measured the flexion of the joints in the thumb, index finger, and pinkie finger. The two tilt sensors measured the tilt of the hand in the horizontal plane, and the two inertial sensors measured the twisting of the forearm and the flexing of the wrist. The drawback of this glove was that it was developed for a specific task and the recognition of hand signs was done strictly in hardware. Therefore, it was not generic enough to perform robust hand posture or gesture recognition in any application other than entry of ASCII characters.

The DataGlove^{TM} and Z-Glove^{TM} developed by VPL Research, were first presented at the Human Factors in Computing Systems and Graphics Interface conference in 1987[121]. Both gloves were designed to be general-purpose interface devices for applications that required direct object manipulation with the hand, finger spelling, evaluation of hand impairment, and the like. Both gloves came equipped with five to fifteen sensors (usually ten) that measured the flexion of both the metacarpophalangeal joints and proximal interphalangeal joints of the four fingers and thumb for a total of 10 DOF. In some cases abduction sensors were used to measure angles between adjacent fingers. Both gloves used optical goniometer sensor technology patented by Zimmerman in 1985. These sensors were made up of flexible tubes with a reflective interior wall, a light source at one end and a photosensitive detector at the other that detected both direct light rays and reflected light rays. Depending on the bending of the tubes, the detector would change its electrical resistance as a function of light intensity[122]. The gloves also provided tactile feedback by putting piezoceramic benders underneath each finger which produced a tingling or numbing sensation. The main difference between the DataGlove and the Z-Glove was the position and orientation mechanisms used with each. A traditional magnetic tracking system was used with the DataGlove, while the Z-Glove had an embedded ultrasonic tracking system that placed two ultrasonic transducers on opposite sides of the metacarpals to measure the roll and yaw of the hand. Generally the Z-Glove was much more limited in application and as a result was less costly.

The DataGlove and Z-Glove were designed as general-purpose interface devices. However, their lack of accuracy limited their utility: formal testing revealed the accuracy of the sensors as no better than five to ten degrees of joint rotation[119]. The gloves could have been used for simple posture recognition and object manipulation, but they were generally not accurate enough for complex gesture recognition.

The Dexterous HandMaster$^{\text{TM}}$ (DHM), first developed in 1987 was an exoskeleton that fit over the hand. Initially it was used as a master controller for the Utah/MIT Dexterous Hand, a four-digit robot hand[67]. A second version of the device later developed and marketed by Marcus[101] used a total of 20 Hall-Effect sensors as potentiometers that measured the flexion of all three joints in each finger, abduction/adduction between each finger, and four degrees of freedom for the thumb. These sensors were sampled at 200 Hz with eight bit accuracy. It was very accurate[2], with a 92 to 98 percent correlation between finger position and DHM readout[64], thus it could have been used for complex posture and gesture recognition, but it took some time to take on and off and was not suited for rapid movements because of its instability when worn.

The Power Glove was developed in 1989 by Mattel as an input device for Nintendo games and, when suitably reverse-engineered for a computer's serial port[30], became a low-cost alternative for researchers in virtual reality and hand posture and gesture recognition[48][85]. The glove used resistive ink sensors that measured the overall flexion of the thumb, index, middle, and ring fingers for a total of four DOF. It also used ultrasonic tracking to track the hand's x, y, and z position and roll orientation of

[2]The device was designed mainly for clinical analysis of hand impairment and robot control.

the wrist relative to a companion unit attached to the display. Because the finger sensors used two bits of precision, the Power Glove was not very accurate and useful only for a small set of simple hand postures and gestures; its big advantage was its extremely low cost.

Finally, the Space Glove™ developed by W Industries in 1991, was unique in that the user placed his fingers and thumb through plastic rings that sat between the proximal interphalangeals and the metacarpophalangeal joints. The glove used sensors with twelve bit analog-to-digital converters that measured the flexion of the metacarpophalangeal joints and the interphalangeal joint of the thumb for a total of six DOF[99]. According to Sturman's personal experience[101], the Space Glove was fairly responsive but uncomfortable to wear due to the inflexibility of the plastic rings around the fingers. The glove worked only with W Industries products and, as a result, little if any work has been done with it.

Current Glove-Based Input Devices One of the least expensive gloves on the market today is the 5DT Data Glove™ (see Figure 2.1) developed by Fifth Dimension Technologies. This glove uses five fiber optic sensors to measure the overall flexion of the four fingers and the thumb; according to the specifications[36], these sensors can be sampled at 200 Hz with eight bits of precision. In addition, the glove uses two tilt sensors to measure the pitch and roll of the hand. The device is currently priced at $495 for a right-handed glove and $535 for a left-handed glove. Since this glove senses only the average flexion of the four fingers and the thumb, it is not suited for complex gesture or posture recognition. However, it does perform well enough for simple postures, such as pointing or making a fist, and is the supported device for General Reality Company's GloveGRASP™ software toolkit for hand posture recognition[39].

The SuperGlove (see Figure 2.2), developed by Nissho Electronics, has a minimum of 10 and a maximum of 16 bend sensors that use a special resistive ink applied to flexible boards sewn into the glove[82]. With its minimal and standard configuration, the SuperGlove measures flexion of both the metacarpophalangeal and proximal interphalangeal joints for all four fingers and the thumb. The glove comes in two different sizes and is available for both the left and right hand. A unique feature of this device is

Figure 2.1: The 5DT Data Glove$^{\text{TM}}$ developed by Fifth Dimension Technologies. The glove measures seven DOF (from Fifth Dimension Technologies[36]).

its embedded calibration procedure: three buttons on the control unit can collect data for three distinct postures to allow hardware calibration. The standard version of the SuperGlove is currently priced at around $5000. A wireless option is also available which increases the price to over $20,000; there are currently no distributors that sell the device in the United States.

From the author's personal experience, the SuperGlove is adequate for simple posture recognition; here a simple posture is a posture having a combination of the fingers either flexed or extended. The glove's hardware-based calibration mechanism is important and does not have to be used often, but it does not remove the need for software calibration. The glove is fairly accurate but not suited for complex gesture recognition. Unfortunately, no formal studies have been performed on the accuracy of the SuperGlove and it is not commonly discussed in the literature.

Pinch$^{\text{TM}}$ Gloves (see Figure 2.3) take a different approach to posture recognition[33]. These gloves, originally called Chord Gloves, were prototyped by Mapes at the University of Central Florida[66]; the technology was bought by Fakespace Inc. and now

Figure 2.2: Nissho Electronics SuperGlove input device. This glove has a minimum of 10 bend sensors and a maximum of 16 (from Nissho Electronics Corporation[82]).

the gloves are sold commercially under the Pinch Glove name. Instead of using bend sensor technology to record joint angles, Pinch Gloves have electrical contacts on the inside of the tips of the four fingers and the thumb. Users can make a variety of postures by completing a conductive path when two or more of the electrical contacts meet. According to Encarnação[31], over 1000 postures are theoretically possible. Usually two gloves are worn to maximize the number of postures available; they are sold in pairs and are priced at $2000/pair.

The Pinch Glove system is excellent for restricted posture recognition because no posture recognition techniques are required (see section 4). The electrical contacts on the gloves make it easy to map postures to a variety of tasks. Since the gloves have a mount for a spatial tracking device such as a Polhemus, simple gestures can also be recognized. The drawbacks of these gloves arise from the fact that they do not use bend sensor technology. It is very difficult to provide a virtual representation of the user's hands, and such a representation is important in virtual environments, although Mutigen's SmartScene has gotten around this by using simple 3D cursors like spheres instead of a virtual hand[76]. Another drawback of Pinch Gloves is that the types of

Figure 2.3: FakeSpace's Pinch^{TM} Glove input devices. The gloves have electrical contact points that allow users to make "pinch" postures that can be then mapped to a variety of tasks.

postures are limited since electrical contacts must be touching before a posture can be recognized. If the user makes a posture in which none of the electrical contacts create a conductive path, the posture goes unrecognized. This type of problem does not occur with a bend-sensor-based glove.

The final glove-based input device discussed here is Virtual Technologies' CyberGlove^{TM} (see Figure 2.4), originally developed by Kramer in his work on The "Talking Glove"[53]. Using his patented strain gauge bend sensor technology[52], he started Virtual Technologies and now sells the glove commercially. The CyberGlove can be equipped with either 18 or 22 bend sensors. With 18 sensors, the CyberGlove measures the flexion of the proximal interphalangeal and the metacarpophalangeal joints of the four fingers and the thumb, the abduction/adduction angles between the fingers, radial and palmer abduction, wrist roll, and wrist pitch[111] (Figure A.2 illustrates the various motions the hand can make). The additional four sensors in the 22 sensor model measure the flexion of the distal interphalangeal joints in the four fingers. With a six DOF tracker and the 22 sensor model, 28 degrees of freedom of the hand can be realized.

An interesting feature of the CyberGlove's interface unit is that it digitizes the voltage output of each sensor and then modifies the value using a linear calibration function. This function uses gain and offset values to represent the slope and y-intercept of the linear equation. This equation allows software calibration of the glove and thus makes it more robust for a variety of hand sizes.

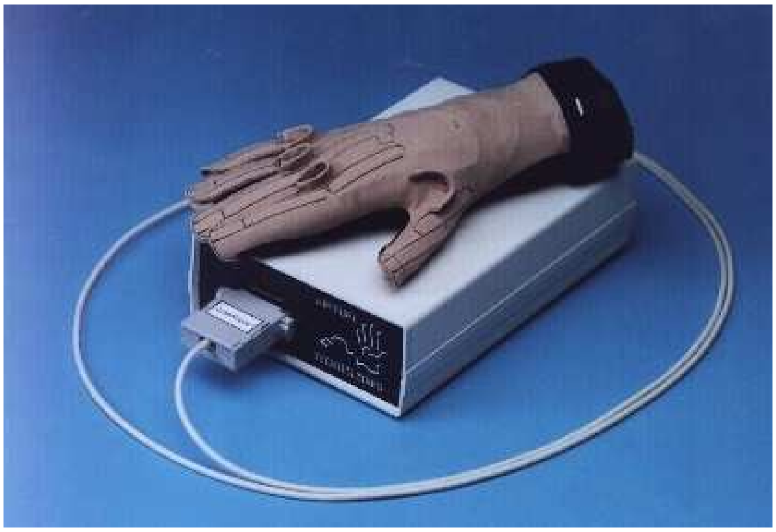

Figure 2.4: Virtual Technologies' CyberGlove and control box. The glove can be equipped with 18 or 22 bend sensors (from Virtual Technologies[110]).

The author's personal experience and an evaluation by Kessler et al. [50] suggest the CyberGlove is accurate to within one degree of flexion. It works well for both simple and complex posture and gesture recognition (Wexelblat[116] and Fels[35] verify this claim). The only negative in regard to the CyberGlove is its price; the 18-sensor model is available for $9800 and the 22-sensor model for $14,500. But even though the glove is expensive, it is the best available glove-based technology for accurate and robust hand posture and gesture recognition.

2.3 Vision-Based Technology

One of the main difficulties in using glove-based input devices to collect raw posture and gesture recognition data is the fact the gloves must be worn by the user and attached to the computer. In many cases, users do not want to wear tracking devices and computer-bound gloves since they can restrict freedom of movement and take considerably longer to set up than traditional interaction methods. As a result, there has been quite a bit of research into using computer vision to track human movement and extract raw data for posture and gesture recognition.

A vision-based solution to collecting data for hand posture and gesture recognition requires four equally important components. The first is the placement and number of cameras used. Placing the camera(s) is critical because the visibility of the hand or hands being tracked must be maximized for robust recognition. Visibility is important because of the many occlusion problems present in vision-based tracking (see section 2.4). The number of cameras used for tracking is another important issue. In general, one camera is used to collect recognition data, and it has been shown by Starner[96] and Martin[68] that one is effective and accurate in recognizing hand posture and gestures. When depth or stereo information is required for tracking hand movement, two or more cameras are needed. Although using more than one camera adds complications due to the algorithmic complexity of dealing with more than one image stream, they do provide more visibility and are critical in virtual environment applications which usually require depth information. Kumo[56] and Utsumi[108] use multiple cameras effectively in the context of 3D object manipulation and 3D scene creation, respectively. Also, Rehag and Kanade[91] have shown that 27 DOF of the hand can be recovered by using two cameras.

The second component in a vision-based solution for hand posture and gesture recognition is to make the hands more visible to the camera for simpler extraction of hand data. One of the first ways of doing this was to place LEDs (light emitting diodes) on various points on the hand[99]. These LEDs let the camera quickly pick up feature points on the hand to aid recognition. A more common method in the literature is to simply use colored gloves. Starner[97], Davis[25], and Kumo[56] have all shown that using solid colored gloves allows faster hand silhouette extraction than simply wearing no gloves at all, but using such gloves makes it difficult to recognize finger movement and bending. In order to achieve fast silhouette extraction and track finger joint movement, Dorner developed a complex encoding scheme using sets of colored rings around the finger joints instead of solid colored gloves[28].

Even though colored gloves are wireless and simple to wear, the ideal situation for vision-based hand tracking is to track the hand with no gloves at all. Tracking a gloveless hand presents some interesting difficulties, among them skin color and background environment issues. In many cases a solid colored screen is placed behind the user so that the natural color of the hands can be found and features extracted. One of the best vision systems for tracking the naked hand was Krueger's VIDEODESK system[55], although it required complex image-processing hardware. He was able to track hand silhouettes in order to create simple 2D and 3D shapes. Utsumi also tracked the naked hand in his 3D scene creation system[108].

The third component of a vision-based solution for hand gesture and posture recognition is the extraction of features from the stream or streams of raw image data; the fourth component is to apply recognition algorithms to these extracted features. Both these components are discussed in section 3.

2.4 Advantages and Disadvantages of Glove- and Vision-Based Data Collection Systems

We can now examine the advantages and disadvantages of glove-based and vision-based technology for hand posture and gesture recognition. Kadous[48] and Sturman[101] have also discussed these issues to varying extents.

Cost Even though glove-based technology has come down in price (under $500 for the 5DT Glove), the cost of robust and complex posture and gesture recognition is going to be high if a glove-based solution is used. The cost of a tracking device and a robust glove is in the thousands of dollars. On the other hand, a vision-based solution is relatively inexpensive, especially since modern-day workstations are equipped with cameras.

User Comfort With a glove-based solution, the user must wear a tracking device and glove that are connected to a computer. Putting these devices on takes time, can be quite cumbersome, and can limit one's range of motion. With a vision-based solution, the user may have to wear a glove, but the glove will be extremely lightweight, easy to put on, and not connected to the computer. Applications in which no gloves are used, give the user complete freedom of motion and provides a cleaner way to interact and perform posture and gesture recognition.

Computing Power Depending on the algorithms used, both glove-based and vision-based solutions can require significant computing power. However, in general, the vision-based approach takes more computing power due to the image processing necessary. Glove-based solutions have a slight advantage over vision-based solutions in that the data the gloves send to the computer can easily be transformed into records that are suitable for recognition. However, with faster computers, computational power should not be an issue.

Hand Size Human hands vary in shape and size. This is a significant problem with glove-based solutions: some users cannot wear these input devices because their hands are too big or too small. This problem is not an issue with vision-based solutions.

Hand Anatomy Glove-based input devices may not always fit well enough to prevent their position sensors from moving relative to the joints the sensors are trying to measure. This problem reduces recognition accuracy after extended periods of use and forces users to recalibrate the devices which can be a nuisance. This problem also is not an issue with vision-based solutions.

Calibration Calibration is important in both vision- and glove-based solutions but, due to the anatomy of the hand, it is more critical with glove-based solutions. In general, a calibration procedure or step is required for every user and, in some cases, every time a user wants to run the system. In some vision-based solutions, however, a general calibration step can be used for a wide variety of users.

Portability In many applications, especially gesture to speech systems, freedom from being tied down to a workstation is important. With glove-based solutions, this freedom is generally available as long as hand tracking is not involved, since these input devices can be plugged right into a laptop computer. Vision-based solutions were originally quite difficult to use in a mobile environment due to camera placement issues and computing power requirements. However, with the advent of wearable computing[65] and powerful laptops with built-in cameras, mobile vision-based solutions are becoming more practical.

Noise In glove-based solutions where hand tracking is required, some type of noise is bound to be associated with the data (it can come from a variety of sources depending on the tracking technology used). Filtering algorithms are therefore necessary to reduce noise and jitter. In some cases this can get computationally expensive when predictive techniques such as Kalman filtering[114] are used. With a vision-based solution, noise from input devices is minimal (although occlusion could be considered a form of noise, since it contributes to the difficulty of vision-based solutions).

Accuracy In both vision- and glove-based solutions for hand posture and gesture recognition, accuracy is one of the most critical components to providing robust recognition. Both these solutions provide the potential for high levels of accuracy depending on the technology and recognition algorithms used. Accuracy also depends on the complexity and quantity of the postures and gestures to be recognized. Obviously, the quantity of possible postures and gestures and their complexity greatly affect accuracy no matter what raw data collection system is used.

3 Algorithmic Techniques for Recognizing Hand Postures and Gestures

Once the raw data has been collected from a vision- or glove-based data collection system, it must be analyzed to determine if any postures or gestures have been recognized. In this section, various algorithmic techniques for recognizing hand postures and gestures are discussed. Although some good surveys have discussed hand gesture and posture recognition techniques[48][97][112], this present survey is considerably more thorough and is up to date with the current literature. The techniques in this survey fall into three rough categories:

- Feature extraction, statistics and models
- Learning algorithms
- Miscellaneous techniques

Each category contains a number of recognition techniques. These techniques will be discussed through a general introduction to the technique, a look at the current literature, and an analysis of advantages and disadvantages.

3.1 Feature Extraction, Statistics, and Models

This category contains six of the most common techniques for hand posture and gesture recognition that extract some mathematical quantity from the raw data. In these cases, the mathematical quantity is represented as a feature, a statistic, or a model (see Table 3.1 for a summary of the techniques).

	Vison	Glove	Postures-Size-Accuracy	Gestures-Size-Accuracy	Training	Previous Work	Adv. Knowledge
Template Matching	Yes	Yes	Complex-Small-98%	Simple-Small-96%	Minimal	Extensive	No
Feature Extraction	No	Yes	Complex-N/A-N/A	Complex-N/A-N/A	None	Moderate	No
Active Shape Models	Yes	No	Simple-Small-N/A	Simple-Small-N/A	None	Minimal	No
Principal Components	Yes	Yes	Complex-Large-99%	N/A-N/A-N/A	Moderate	Moderate	No
Linear Fingertip Models	Yes	No	Complex-Small-90%	N/A-N/A-N/A	Minimal	Minimal	No
Causal Analysis	Yes	No	N/A-N/A-N/A	Simple-Small-N/A	Minimal	Minimal	No

Table 3.1: A summary of the hand posture and gesture recognition techniques found in Section 3.1. The table shows information about whether a technique has been used in a glove- or vision-based solution, the posture and gesture complexity, set size, and reported accuracy, the extent of the training required, how much work has been done using the technique, and if it is important to have advance knowledge of the posture or gesture set during implementation.

3.1.1 Template Matching

Template matching is one of the simplest methods for recognizing hand postures and has been discussed frequently, with thorough contributions by Sturman[101] and Watson[112]. Templates can be used in both glove-based and vision-based solutions; the templates are sequences of sensor values (gloves) and a static or small set of images (computer vision). Here we discuss only gloved-based template matching although it is used in vision-based solutions as well.

In general, template matching determines whether a given data record can be classified as a member of a set of stored data records. Recognizing hand postures using template matching has two parts. The first is to create the templates by collecting data values for each posture in the posture set. Generally, each posture is performed a number of times and the average of the raw data for each sensor is taken and stored as the template. The second part is to compare the current sensor readings with the given set of templates to find the posture template most closely matching the current data record.

An example of the template matching comparison is the use of a Boolean function on the results of the explicit comparison between each sensor value in the current data record and the corresponding value in the posture templates. The comparisons are often made within a range of values, which helps to improve recognition accuracy with noisy glove devices. A problem with comparisons of angle measurements within a range of values, however, is that while in theory, these ranges usually go from zero to a power of two based on the bits of precision in each bend sensor, the actual angle measurements from the bend sensors do not follow their theoretical ranges and each bend sensor often has a different range. A way to remedy this problem is to normalize the bend sensor's measurements using the maximum and minimum value for each bend sensor. Normalization makes all the angle measurements zero for full extension and one for full flexion, thus making comparisons with ranges easier to implement. The drawback of normalizing bend angles is that maximum and minimum values can change during glove use; however dynamic calculation and updating of maximum and minimum values has been shown to combat this problem[101].

Another example of template matching comparisons is the use of distance measurements between the current data record and each of the data records in the posture set recognizing the posture with the lowest distance measurement. The distance measurement must be below some threshold value to avoid false positive recognition. Two distance measurements used for hand posture template matching are the sum of the absolute differences[121] and the sum of the squares[81]. The main advantage of computing a distance measurement is that comparison ranges need not be used. The main disadvantage is that a measurement must be made for each posture template.

Template matching is the simplest of the hand posture and gesture recognition techniques, and for a small set of postures, it is appropriate and can be quite accurate. But the technique does have limitations. First, template matching is much more difficult for hand gestures. However, Darrell and Pentland have recognized hand gestures in a vision-based solution using sets of view models or templates that are matched with gesture patterns using dynamic time warping[27][1]. The second limitation is the small number of possible postures that can be recognized. If the application requires a large posture set, then template matching will not work since the posture templates will overlap[112].

[1] In this case, two gestures were recognized[27].

Strengths

I. **Simplest technique to implement**

II. **Accurate (for small set of postures)**

III. **Requires only a small amount of calibration**

Weaknesses

I. **Not suited for hand gestures**

II. **Does not work well for large posture sets due to overlapping templates[112]**

3.1.2 Feature Extraction and Analysis

In feature extraction and analysis, low-level information from the raw data is analyzed to produce higher-level semantic information and then used to recognize postures and gestures. One of the first gestural interfaces to use a feature-based system was Rubine's 2D single-stroke gesture recognizer[92]. Rubine extracted such features as the cosine and sine of the initial angle of the gesture, the distance between the first and last point, the maximum speed of the gesture, and so on. From these features, the system recognized gestures that represented numbers and letters of the alphabet, among others. The system recognized these gestures with over 97% accuracy.

This type of feature-based approach has also been applied to recognizing hand postures and gestures. However, it is slightly more complex due to the increase in dimensions from two to three and the increase in the amount of raw data produced by the input devices. Sturman[101] was the first person to extend the ideas behind Rubine's work into three dimensions and to make possible continual analysis and recognition (instead of requiring a starting and ending point in the gesture). Sturman used explicit formulations for each gesture that do not require training by individual users; however, manual programmer intervention is necessary to add new gestures. Both position data for a tracker and flex-sensor data were kept for feature extraction. The features used were similar to Rubine's but also included such inherently 3D features as cross product and bounding volume.

Using the work by Rubine and Sturman, Wexelblat developed a robust hand gesture and posture analyzer useful in a variety of applications[116]. Wexelblat's system uses a hierarchical structure that moves raw data from two CyberGloves and a set of trackers through a series of layers, each layer analyzing the data to provide higher level semantic meaning[117]. The first layer above the input devices uses entities called segmenters to watch the raw data. The segmenters look for significant changes over time in the raw data values. Once a significant change is found, the segmenter sends a data record consisting of information that represents the data change to the next layer in the analyzer.

The next layer in Wexelblat's analyzer has a set of proto-feature detectors that extract (from the data records of one or more segmenters) information such as the extension of a finger or curvature in the palm. This information is sent to the higher-level feature detectors, which use the information from one or more proto-feature detectors to derive more global features like a fist, flat hand posture or a waving gesture. The path analysis module then takes data from all the feature detectors and puts the information into a frame to hold the completed description of the gesture. Finally, the frames are sent to an integration module that handles the interaction and performs temporal integration over them.

Feature extraction and analysis is a robust way to recognize hand postures and gestures. It can be used not only to recognize simple hand postures and gestures but complex ones as well. Its main drawback is that it can become computationally expensive when a large number of features are being collected, which slows down system response time. This slowdown is extremely significant in virtual environment applications, but should diminish with the advent of faster machines. Finally, note that Wexelblat's feature extraction and analysis method could be used in vision-based solutions by altering how the features are extracted. With a glove-based solution, features are extracted from bend sensor and tracker data, while a vision-based solution would require features to be extracted from images.

Strengths

I. Handles postures and gestures equally well

II. **Uses layered architecture to analyze postures and gestures**

Weaknesses

I. **Can be computationally expensive depending on how many features are extracted**

3.1.3 Active Shape Models

Active shape models, or "smart snakes" as they are sometimes called, are a technique for locating a feature within a still image[23]. The technique places a contour on the image that is roughly the shape of the feature to be tracked. The contour is then manipulated by moving it iteratively toward nearby edges that deform the contour to fit the feature. Heap and Samaria extend this technique to recognize hand postures and gestures using computer vision[44]. In their system, they apply an active shape model to each frame and use the position of the feature in that frame as an initial approximation for the next frame. They also use a point distribution model[22] to find a suitable model for the tracked object and the inherent application specificity of tracking a human hand.

Heap and Samaria's system runs in real time (25 frames per second) and is applicable only to vision-based solutions. The main disadvantage of this technique is that currently it can track only the open hand, which severely limits the number of hand postures and gestures that can be recognized. Also, there is very little empirical evidence in the literature to support its validity. Open areas of research using active shape models include extending them to the 3D domain (i.e. using multiple cameras) so that the number of possible postures and gestures can be increased, and determining the accuracy of the technique.

Strengths

I. **Allows real time recognition**

II. **Handles both hand postures and gestures**

Weaknesses

I. **Tracks only the open hand**

II. Has not been applied to stereo data from multiple cameras

3.1.4 Principal Component Analysis

Principal component analysis (PCA) is a statistical technique for reducing the dimensionality of a data set in which there are many interrelated variables, while retaining as much of the variation in the dataset as possible[47]. The data set is reduced by transforming the old data to a new set of variables (principal components) that are ordered so that the first few variables contain most of the variation present in the original variables. The original data set is transformed by computing the eigenvectors and eigenvalues of the data set's covariance matrix. The eigenvector with the highest eigenvalue holds the highest variance, the eigenvector with the second highest eigenvalue holds the second highest variance, and so on.

PCA was first applied in the computer vision community to face recognition by Sirovich and Kirby[95] and later extended by Turk and Pentland[106]. Birk et al. and Martin independently developed the first two systems using PCA to recognize hand postures and gestures in a vision-based system[11][68]. Birk's system was able to recognize 25 postures from the International Hand Alphabet, while Martin's system was used to interact in a virtual workspace.

Birk's system first performs PCA on sets of training images to generate a posture classifier that is then used to classify postures in real time. Each set of training images can be considered a multivariate data set: each image consists of N pixels and represents a point in N-dimensional space. In order for PCA to work successfully, there must be little variance in at least one direction and whatever variance exists should not be meaningful. Birk's recognition system works well but there is little indication that PCA compresses the data set significantly beyond a naive approach.

Another important issue when dealing with image data is that it is highly sensitive to position, orientation, and scaling of the hand in the image. PCA cannot transform two identical postures with different hand sizes and positions to the same point. Birk thus normalizes each image to center the hand, rotate it so that its major axis is vertical, and scale it to fit the gesture image.

The posture classifier, which is created off line, is a transformation matrix containing the results of the PCA performed on all the images in the training set. After the transformation matrix has been calculated, the number of principal components is reduced by discarding the least important ones. The common approach is simply to keep the principal components with the *n* highest eigenvalues. However, Birk has shown that this is not effective when only a small number of principal components are to be kept[12]. Other information such as the number of posture classes, their mean, and their variance in the reduced data set can be used to choose the principal components. A Bayes classifier is then used to recognize postures from the reduced set of principal components. Note that such parameters as image resolution and number of training images are also important components of PCA in a vision-based solution and can be modified to improve recognition results[11].

Although principal component analysis can be used in a glove-based solution[105], it has been used primarily in the computer vision community. It is accurate for specific posture sets such as the International Hand Alphabet[11] but requires training by more than one person to provide robust results. More research still needs to be done to measure the validity of this technique and to determine whether more complex hand gestures can be recognized accurately.

Strengths

- **I. Can recognize on the order of 25 to 35 postures[11][105]**

Weaknesses

- **I. Requires training by more than one person for accurate results and user independence[11]**
- **II. Requires normalization to keep images consistent**

3.1.5 Linear Fingertip Models

The linear fingertip model assumes that most finger movements are linear and comprise very little rotational movement. This assumption allows for a simplified hand model that uses only the fingertips as input data and permits a model that represents each fingertip trajectory through space as a simple vector. Davis and Shah use this approach in a vision-based solution that puts a glove with brightly colored fingertips on the user's hand[25] and extracts the fingertip positions using histogram segmentation[40]. Once the fingertips are detected, their trajectories are calculated using motion correspondence[90]. The postures themselves are modeled from a small training set by storing a motion code, the gesture name, and direction and magnitude vectors for each of the fingertips. The postures are recognized if all the direction and magnitude vectors match (within some threshold) a gesture record in the training set. System testing showed good recognition accuracy (greater than 90%), but the system did not run in real time and the posture and gesture set should be expanded to determine if the technique is robust.

Strengths

I. **Simple approach**

II. **Concerned only with starting and ending points of fingertips**

III. **Has good recognition accuracy**

Weaknesses

I. **System did not run in real time**[2]

II. **Recognizes only a small set of postures**

III. **Does not take curvilinear fingertip motion into account**

3.1.6 Causal Analysis

Causal analysis is a vision-based recognition technique that stems from work in scene analysis[16]. The technique attempts to extract information from a video stream by using high-level knowledge about actions in the scene and how they relate to one another and the physical environment. Examples of causal data (e.g. the underlying physics of the scene) include rigidity, mass, friction, balance, and work against gravity. Brand

[2] With today's computer performance, this system should run in real time.

uses this information in his system, BUSTER, that understands and outputs information about the structure and stability of block towers[16]; In [15], Brand provides a more detailed description on BUSTER and other scene analysis systems.

Brand and Essa have applied causal analysis to vision-based gesture recognition[14]. By using knowledge about body kinematics and dynamics, features recovered from the video stream can be used to identify gestures based on human motor plans. The system captures information on shoulder, elbow and wrist joint positions in the image plane. From these positions, the system extracts a feature set that includes wrist acceleration and deceleration, work done against gravity, size of gesture, area between arms, angle between forearms, nearness to body, and verticality. Gesture filters normalize and combine the features and use causal knowledge of how humans interact with objects in the physical world to recognize gestures such as opening, lifting, patting, pushing, stopping, and clutching.

Causal analysis in gesture recognition is an interesting concept, but Brand and Essa's discussion of their implementation is cursory[14] and, as a result, it is unclear how accurate their system is. This system also has the disadvantage of not using data from the fingers. More research needs to be conducted in order to determine if this technique is robust enough to be used in any nontrivial applications.

Strengths

- **I. Uses information about how humans interact with the physical world to help identify gestures**

Weaknesses

- **I. Uses only a limited set of gestures**
- **II. Does not use hand orientation and position data or finger data**
- **III. Does not run in real time**
- **IV. Implementation is unclear[14].**

3.2 Learning Algorithms

Here we describe the use of three of the most common learning algorithms used to recognize hand postures and gestures. These algorithms all stem from the artificial intelligence community, and their common trait is that recognition accuracy can be increased through training (see Table 3.2 for a summary of these techniques).

	Vison	Glove	Postures-Size-Accuracy	Gestures-Size-Accuracy	Training	Previous Work	Adv. Knowledge
Neural Networks	Yes	Yes	Complex-Large-98%	Complex-Small-96%	Extensive	Extensive	Yes
Hidden Markov Models	Yes	Yes	Complex-Large-90%	Complex-Small-96%	Extensive	Extensive	Yes
Instance-based Learning	No	Yes	Complex-Large-80%	N/A-N/A-N/A	Extensive	Minimal	No

Table 3.2: A summary of the hand posture and gesture recognition techniques found in Section 3.2. The table shows information about whether a technique has been used in a glove- or vision-based solution, the posture and gesture complexity, set size, and reported accuracy, the extent of the training required, how much work has been done using the technique, and if it is important to have advance knowledge of the posture or gesture set during implementation.

3.2.1 Neural Networks

This section presents a brief introduction into the concepts involved in neural networks. For a more comprehensive description, see Russell and Norvig[93], Krose and van der Smagt[54] or Anderson[5].

A neural network is an information processing system loosely based on the operation of neurons in the brain. While the neuron acts as the fundamental functional unit of the brain, the neural network uses the node as its fundamental unit; the nodes are connected by links, and the links have an associated weight that can act as a storage mechanism[93]. Each node is considered a single computational unit containing two components. The first component is the input function which computes the weighted sum of its input values; the second is the activation function, which transforms the weighted sum into a final output value. Many different activation functions can be used; the step, sign, and sigmoid functions being quite common[93] since they are all relatively simple to use. For example, using the step function, if the weighted sum is above a certain threshold, the function outputs a one indicating the node has "fired" otherwise it outputs a zero indicating the node has not fired. The other two activation functions act in a similar manner.

Neural networks generally have two basic structures or topologies, a feed-forward structure and a recurrent structure. A feed-forward network can be considered a directed acyclic graph, while a recurrent network has an arbitrary topology. The recurrent network has the advantage over a feed-forward network in that it can model systems with state transitions. However, recurrent networks require more complex mathematical descriptions and can exhibit chaotic behavior. In both network topologies, there is no restriction on the number of layers in the network. These multilayered networks provide more representation power at the cost of more complex training. The nodes in between the input and output nodes of the multilayered network have no communication with the outside world and cannot be directly observed from the input or output behavior of the network. If the number of hidden nodes is large, it is possible to represent any continuous function of the inputs[93].

Training is an important issue in neural networks and can be classified in two different ways. First, supervised learning trains the network by providing matching input and output patterns; this trains the network in advance and as a result the network does not learn while it is running. The second learning mechanism is unsupervised learning or self-organization which trains the network to respond to clusters of patterns within the input. There is no training in advance and the system must develop its own representation of the input, since no matching output is provided[54]. Note that supervised and unsupervised learning do not have to be mutually exclusive: depending on the network, a combination of the two learning strategies can be employed. Neural network training is one of most important areas of research in neural network technology, but the many different algorithms for supervised and unsupervised learning strategies are beyond the scope of this survey; for a thorough discussion see Mehrotra, Mohan, and Ranka[71].

Neural networks have been used principally in the artificial intelligence community to build certain types of autonomous agents and recognize patterns. One of the first systems to use neural networks in hand posture and gesture recognition was developed by Murakami[77]. Hand postures were recognized with a three-layered neural network that contained 13 input nodes, 100 hidden nodes, and 42 output nodes, one for each posture to be recognized. The network used back propagation, a learning mechanism that minimizes the error between target output and the output produced by the network[93], and achieved 77% accuracy with an initial training set. Accuracy increased to 98% for participants in the original training set when the number of training patterns was increased from 42 to 206. Hand gesture recognition was done with a recurrent three-layered network with 16 input units, 150 hidden units, and 10 output units, one for each of the 10 possible gestures recognized. Recurrency in the network allowed for processing of time variant data. Recognition accuracy was initially 80%, but 96% recognition rates were achieved with a filtering mechanism for the raw data[3].

Fels did extensive work with neural networks with his Glove-TalkII system[34] in which three back propagation neural networks are used to translate hand gestures into speech. The hand gesture recognition process was broken up into three networks in order to increase speed and reduce training time. The first network, the vowel/consonant network, determined if the user wanted to produce a vowel or a consonant. This network employed a feed-forward topology with 12 input nodes, 10 hidden nodes and one output node. The second network was used to generate consonant sounds. It also employed a feed-forward topology but used 12 input nodes, 15 hidden nodes, and nine output nodes. The third network used to generate vowel sounds, had a feed-forward structure and employed two input nodes, 11 hidden nodes, and eight output nodes. The consonant and vowel networks used normalized radial basis activation functions[17] for the hidden inputs that solved problems arising from similar-sounding consonants and vowels (see Fels[35]). With these networks, a well-trained user (100 hours of training, including training the networks) was able to make intelligible speech that sounded somewhat natural.

Another system using neural networks developed by Banarse[8] was vision-based and recognized hand postures using a neocognitron network, a neural network based on the spatial recognition system of the visual cortex of the brain (for a detailed de-

[3]See Murakami for details on this mechanism[77].

scription of the neocognitron see Fukushima[38]). However, the system was limited and recognized only a handful of postures. More extensive research is needed to determine the validity of the neocognitron network as a hand posture and gesture recognition technique.

Neural networks are a useful method for recognizing hand postures and gestures, yield increased accuracy conditioned upon network training, and work for both glove-based and vision-based solutions. However, they have distinct disadvantages. First, different configurations of a given network can give very different results, and it is difficult to determine which configuration is best without implementing them: Fels[35] reports that he implemented many different network configurations before finding ones that provided good results. Another disadvantage is the considerable time involved in training the network. Finally, the whole network must be retrained in order to incorporate a new posture or gesture. If the posture or gesture set is known beforehand this is not an issue, but if postures and gestures are likely to change dynamically as the system develops, a neural network is probably not appropriate.

Strengths

- **I. Can be used in either a vision- or glove-based solution**
- **II. Can recognize large posture or gesture sets**
- **III. With adequate training, high accuracy can be achieved**

Weaknesses

- **I. Network training can be very time consuming and does not guarantee good results**
- **II. Requires retraining of the entire network if hand postures or gestures are added or removed**

3.2.2 Hidden Markov Models

We first give a brief introduction of the concepts involved in hidden Markov models. For a more detailed description see Rabiner and Juang[88], Rabiner[89], Huang et al. [45], or Charniak[21].

In describing hidden Markov models it is convenient first to consider Markov chains. Markov chains are simply finite-state automata in which each state transition arc has an associated probability value; the probability values of the arcs leaving a single state sum to one. Markov chains impose the restriction on the finite-state automaton that a state can have only one transition arc with a given output; a restriction that makes Markov chains deterministic. A hidden Markov model (HMM) can be considered a generalization of a Markov chain without this Markov-chain restriction[21]. Since HMMs can have more than one arc with the same output symbol, they are non-deterministic, and it is impossible to directly determine the state sequence for a set of inputs simply by looking at the output (hence the "hidden" in "hidden Markov model").

More formally, a HMM is defined as a set of states of which one state is the initial state, a set of output symbols, and a set of state transitions. Each state transition is represented by the state from which the transition starts, the state to which transition moves, the output symbol generated, and the probability that the transition is taken[21]. In the context of hand gesture recognition, each state could represent a set of possible hand positions. The state transitions represent the probability that a certain hand position transitions into another; the corresponding output symbol represents a specific posture and a sequence of output symbols represent a hand gesture. One then uses a group of HMMs, one for each gesture, and runs a sequence of input data through each HMM. The input data, derived from pixels in a vision-based solution or from bend sensor values in a glove-based solution, can be represented in many different ways, the most common by feature vectors[97]. The HMM with the highest forward probability (described later in this section) determines the users' most likely gesture. An HMM can also be used for hand posture recognition; see Liang and Ouhyoung[61] for details.

A number of important issues arise in dealing with HMMs. As with neural networks, training HMMs is very important for increasing recognition accuracy of the model. A common approach is to adjust the HMM's transition probabilities in order to optimize it for a training data set. If the training data is an accurate representation of a particular gesture, for example, then the HMM should be able to recognize that gesture given new data. The Baum-Welch algorithm[21][88][89] uses the given training sequence to reestimate the probabilities of the state transitions in the HMM.

One of the components of the Baum-Welch algorithm is forward probability. Forward probability, or alpha probability as it is sometimes called, is the probability of an HMM given an output sequence and is calculated by incrementally computing the probabilities for the output on a symbol-by-symbol basis[88]. The algorithm goes through each timestep t and examines each state j in the HMM. For each state j, it computes a summation of the probability of producing the current output symbol at t and moving to j given that the algorithm was in state k, multiplied by the probability of producing the output up to t and ending up in k. Note that the summation is over all k. Performing this calculation iteratively for each output symbol yields the probability that the HMM would generate the whole output sequence. The forward probability is used to find a HMM with the highest probability of matching an output sequence. For example, if a hand gesture set contains 10 gestures, each one having its own HMM, the HMM with the highest forward probability score would determine which gesture to recognize. See Charniak[21] for a more detailed description of forward probability.

As stated previously, one of the drawbacks of HMMs is that one cannot directly determine the state sequence for a set of output symbols since a state can have more than one arc with the same output. Nevertheless, this information can be approximated by finding the most likely states in the HMM. A common technique for finding the best state sequence for a given output, the Viterbi algorithm[88], uses calculations similar to forward probability except the maximum is taken instead of taking a summation over all k. The Viterbi algorithm is useful because it is a fast method for evaluating HMMs. For more detail on the Viterbi algorithm see Charniak[21].

Hidden Markov models were first used in the recognition community for recognizing handwriting[109] and speech[45], and more recently a significant amount of research has been done on applying HMMs to hand posture and gesture recognition. Starner used HMMs in a vision-based solution to recognize a forty word subset of American Sign Language[97]. Instead of using a different model for each sign in the recognition set, Starner found the minimum and maximum number of states required for an individual HMM and then, using skip transitions (which give low weights to state transitions that are not needed) developed a general HMM topology for all models used in the system. With ample training of the HMMs (between 20 and 80 training samples for each sign) the system was able to achieve an accuracy of over 90 percent.

Schlenzig also used a single HMM to recognize hand gestures in a vision-based solution[94]. The state of the HMM represents the gestures and the observation symbols represent the current static hand posture. This HMM had three possible states and nine possible observation symbols so the number of recognizable gestures was limited. The system employs a recursive filter for updating estimates of the gesture being recognized based on the current posture information. This recursive estimator allows recognition of combined gestures and requires only one HMM.

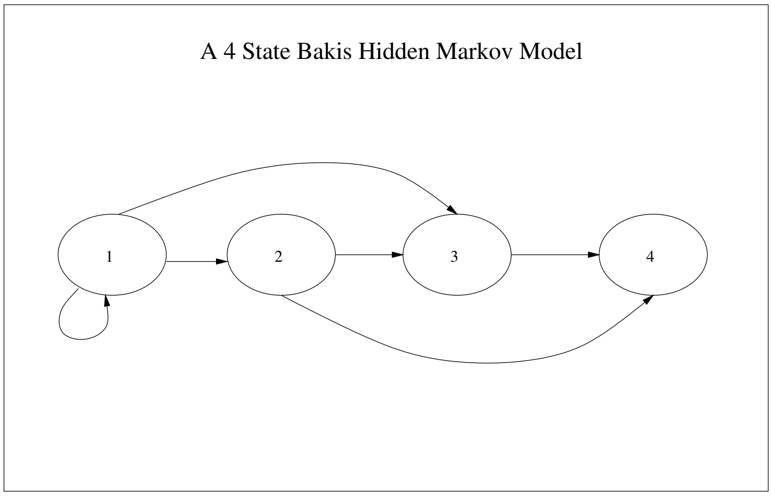

Figure 3.1: A four state Bakis HMM.

Lee and Xu were able to recognize a set of 14 gestures in the context of robot teleoperation[60]. The system uses a CyberGlove and implements a Bakis hidden Markov model that restricts state transitions so that a given state can have a transition only to itself or one of the next two states; for example, if the HMM has four states, state two could have a transition arc only to itself, state three, or state four (see Figure 3.1). This type of model assumes that the recognizable gestures have no cyclical properties. The system also performed iterative model updates as each gesture is recognized. Recognition accuracy of over 90 percent was achieved.

Nam and Wohn[79] and Liang and Ouhyoung[61] have also explored the use of HMMs in hand posture and gesture recognition. Nam and Wohn use a glove-based solution with a VPL DataGlove and Polhemus tracker and reduce the dimensional complexity of the tracking device from 6D (three position axes and three orientation axes) to 2D by using plane fitting. The 2D information is then sent to the HMM. After training each of the 10 HMMs, one for each gesture, with approximately 200 training samples per gesture, accuracy of over 96 percent was attained. Liang and Ouhyoung use HMMs to recognize 50 signs in Taiwanese Sign Language. The system uses a *n*-best approach to outputting recognition results. Unfortunately, no recognition accuracies were reported.

Hidden Markov models provide a good way to perform hand posture and gesture recognition, and can be used in both vision-based and glove-based solutions. The literature has shown that high accuracy can be achieved and the number of possible hand gestures or postures in a posture or gesture set can be quite large. Like neural networks, HMMs must be trained and the correct number of states for each posture or gesture must be determined to maximize performance. If the number and types of hand posture and gestures are known beforehand, HMMs are a good choice for recognition. If the hand postures and gestures are determined as the system is developed, the development process can be more time-consuming due to retraining. If one HMM is used for all gestures, as in Starner's work, then the single HMM must be retrained. If each gesture has an associated HMM, then only the new gesture's HMM will have to be trained. Although HMMs require extensive training, and their hidden nature makes it difficult to understand what is occurring within them, they still may be the technique of choice since they are well covered in the literature and the accuracies reported are usually above 90 percent.

Strengths

- **I. Can be used in either a vision- or glove-based solution**
- **II. Can recognize large posture or gesture sets**
- **III. With adequate training, high accuracy can be achieved**
- **IV. Well discussed in the literature**

Weaknesses

I. Training can be time consuming and does not guarantee good results

II. As with multi-level neural networks, the hidden nature of HMMs makes it difficult to observe their internal behavior

3.2.3 Instance-Based Learning

Instance-based learning is another recognition technique that stems from work done in machine learning. The main difference between instance-based learning and other learning algorithms such as neural networks and hidden Markov models is the way in which the training data is used. With supervised neural networks, for example, the training data is passed through the network and the weights at various nodes are updated to fit the training set. With instance-based learning, the training data is simply used as a database in which to classify other "instances". An instance, in general, is a vector of features of the entity to be classified. For instance, in posture and gesture recognition, a feature vector might be the position and orientation of the hand and the bend values for each of the fingers.

Instance-based learning methods include techniques that represent instances as points in Euclidean space, such as the K-Nearest Neighbor algorithm, and techniques in which instances have a more symbolic representation, such as case-based reasoning[74]. In the K-Nearest Neighbor algorithm, an instance is a feature vector of size n with points in n-dimensional space. The training phase of the algorithm involves storing a set of representative instances in a list of training examples. For each new record, the Euclidean distance is computed from each instance in the training example list, and the K closest instances to the new instance are returned. The new instance is then classified and added to the training example list so that training can be continuous. In the case of hand posture recognition, the training set would be divided into a number of categories based on the types of recognizable postures. As a new posture instance is entered, its K nearest neighbors are found and used to determine the category in which the instance should be placed (thus recognizing the instance as a particular posture).

Another type of instance-based learning technique is case-based reasoning, in which instances have more elaborate descriptions. A typical instance in a case-based reasoning system might be a description of a mechanical part. Since the instance descriptions are more elaborate, simple approaches to determining instance similarity, such as the Euclidean distance measure, are not applicable. Other methods for determining similarity between instances must be used, such as those found in knowledge base techniques. See Mitchell for more detail on case-based reasoning, the K-Nearest Neighbor algorithm, and other instance-based learning algorithms[74].

Instance-based learning techniques have the advantage of simplicity, but they have a number of disadvantages as well. One major disadvantage is the cost of classifying new instances. All of the computation must be done whenever a new instance is classified, which means there will be response time issues when dealing with a large amount of training examples. Another disadvantage of these methods is that not all of the training examples may fit in main memory, and thus will also increase response time. Note that Aha has developed two instance-based learning algorithms that alleviate some of these problems[2]. The first one saves space by discarding instances that have already been correctly classified, and the second makes assumptions about the data to weed out irrelevant instances.

Unfortunately, very little work has been done on instance-based learning in recognizing hand postures and gestures. One of the few systems reported in the literature was developed by Kadous[48], which recognized 95 discrete hand postures for performing sign language recognition using the three instance-based learning algorithms described by Aha[2]. An interesting feature of this system was its capability of achieving approximately 80% accuracy with the use of a Power Glove as the raw data collection unit.

Instance based learning shows promise as a way to recognize hand postures. However, response time may be an important factor when issuing posture commands due to the amount of computation required when each instance is generated, especially when instances are generated at 30 or more per second (based on the speed of the input devices used). More research is needed to determine whether the technique can be applied to hand gestures and if the accuracy can be improved.

Strengths

I. **Except for case-based reasoning, instance-based learning techniques are relatively simple to implement**

II. **Can recognize a large set of hand postures with moderately high accuracy**

III. **Provides continuous training**

Weaknesses

I. **Requires a large amount of primary memory as training set increases**

II. **Response time issues may arise due to a large amount of computation at instance classification time**

III. **Only a little reported in the literature on using instance-based learning with hand postures and gestures**

3.3 Miscellaneous Techniques

This section presents three other techniques, for recognizing hand gestures and postures: the linguistic approach, appearance-based motion analysis and spatio-temporal vector analysis (see Table 3.3 for a summary of these techniques).

	Vison	Glove	Postures-Size-Accuracy	Gestures-Size-Accuracy	Training	Previous Work	Adv. Knowledge
Linguistic Approach	Yes	Yes	Simple-Small-50%	N/A-N/A-N/A	N/A	Minimal	No
Appearance-based Motion	Yes	No	N/A-N/A-N/A	N/A-N/A-N/A	N/A	Minimal	No
Spatio Temporal Vectors	Yes	No	Complex-Medium-N/A	Complex-Medium-N/A	Minimal	Minimal	No

Table 3.3: A summary of the hand posture and gesture recognition techniques found in Section 3.3. The table shows information about whether a technique has been used in a glove- or vision-based solution, the posture and gesture complexity, set size, and reported accuracy, the extent of the training required, how much work has been done using the technique, and if it is important to have advance knowledge of the posture or gesture set during implementation.

3.3.1 The Linguistic Approach

The linguistic approach uses a formal grammar to represent the hand posture and gesture set. Hand[43] used this approach to recognize a small set of postures under the constraint that postures have fingers either fully flexed or fully extended in a number of configurations. Hand postures were mapped to the grammar, which was specified by a series of tokens and production rules. The system used a Power Glove and an ultrasonic tracker to record raw data. Recognition accuracy was poor, ranging from about 15 to 75 percent depending on the posture to be recognized. Hand[43] claims the poor recognition rate may be due to the inaccuracy of the Power Glove; however, Kadous's work[48] with a Power Glove for instance-based learning (see section 3.2.3) achieved a much higher recognition rate using a larger posture set.

Strengths

- **I. Simple approach**
- **II. Can be used in either a vision- or glove-based solution**

Weaknesses

- **I. Poor recognition results**
- **II. Limited to only simple hand postures**
- **III. Little work reported in the literature using this technique to recognize hand postures and gestures**

3.3.2 Appearance-Based Motion Analysis

Appearance-based motion analysis exploits the observation that humans can recognize actions from extremely low resolution images and with little or no information about the three-dimensional structure of the scene. Using this observation, Davis[25] developed an appearance-based recognition strategy for human motion using a hypothesize and test paradigm[42]. This strategy recognizes motion patterns from a video stream by first creating a filter called a binary motion region (BMR), which describes the spatial distribution of motion energy for a given action. In other words, the filter highlights regions in an image in which any form of motion has been present since the beginning of the action. The BMR acts as an index into a database of binary motion regions collected from training sessions. If the current BMR is close to any of the stored BMRs, then the current BMR is tested against a motion model of an action.

Once a set of possible actions has been found by the current BMR, the unknown movement is classified as one of the predefined actions. Davis developed two approaches to represent actions for classification. The first approach creates specific region-based motion parameterizations by reducing motion time traces of particular regions of the image to a single coefficient vector where statistics are used for classification. The second approach breaks down the motion information of an action sequence into a single image based on a motion history image that is calculated using the pixel intensity as a function of the motion history at a given location. Features from the single image are extracted and matched (using the Mahalanobis distance[4] metric[32]) against known movement[5].

The system was tested for a small set of motions (sitting, arm waving, and crouching) with good results (90%) for those subjects included in the training phase. Subjects who were not included in the training phase performed significantly lower (76%) which Davis claims is representative of not enough training data. Although it did not specifically recognize hand gestures, appearance-based motion analysis could be used to recognize very simple ones, but the technique will not work with gestures and postures that have complicated finger movement. More research is needed to determine if a more complex set of motions can be recognized with the existing technique, since most hand gestures are more complicated than the set of test motions used in Davis's evaluation.

Strengths

I. Provides unobtrusive recognition

[4]The Mahalanobis distance contructs an ellipsoid in multidimensional space where variations in the directions of the shorter axes have more weight.

[5]Note that a detailed description of the recognition technique described in this and the previous paragraph can be found in Davis[25].

II. Accurate recognition with a small set of motions for the trained user

Weaknesses

I. Has not been used for recognizing hand posture and gestures

II. Very difficult to detect small details in finger movement

3.3.3 Spatio-Temporal Vector Analysis

Spatio-temporal vector analysis is used to track the boundaries of the moving hand in a vision-based system[87]. This technique makes it possible to extract the image flow from a hand gesture which then can be used for hand gesture interpretation and recognition. Quek[87] computes the spatio-temporal vectors in three steps. First, the images are processed to find the location of moving edges; second, an initial flow field describing the velocities at moving edge points is computed; finally, the flow field is refined and smoothed using a variance constraint.

The location of moving edges is calculated from the assumption that the moving hand is the fastest moving object in the static scene. The video image is modeled using a three-dimensional signal specifying an *xy* point and time. Partial derivatives with a varying *x* and *y* are computed from this signal using Sobel operators (a commonly used technique in image processing). The Sobel operators yield gradient images in which the image value is a measure of pixel intensity. These images are then combined with their corresponding time-derivative images to yield new images in which the pixel intensity is a measure of moving-edge intensity. The edges are extracted from these derived images by eliminating edge points whose neighbors parallel to the edge direction do not have a maximum magnitude.

After the moving edges have been found, the vector flow field of moving edges is computed. Edge velocities are calculated by first choosing dominant edge points at which to calculate velocities. Then edge point correspondences from different edge images are found by a correlation process called absolute difference correlation[1]. After the correlation process is complete, the best set of vectors from the endpoint candidates is found. This set is calculated with a variance constraint that represents the normalized velocity change across the image's vector field. The best set of vectors is then used to recognize the predefined hand gestures[6].

More research needs to be conducted with concrete accuracy testing to determine if spatio-temporal vector analysis can recognize hand postures and gestures adequately.

Strengths

I. **Provides unobtrusive recognition**

II. **Can recognize a medium-sized set of hand postures and gestures**

Weaknesses

I. **No recognition accuracy results reported**

II. **Requires significant computation to track the hands**

[6] For details on the mathematics behind this technique see Quek[87].

4 Applications Areas That Use Hand Postures and Gestures

This section discusses the various applications that have used hand postures and gestures as their interaction metaphor. Note that Sturman and Zeltzer[99] and Sturman[101] have also reported on application areas that use hand postures and gestures. Unfortunately, in many of these application areas, usability and recognition accuracy measures have not been reported.

4.1 Sign Language

One of the more intuitive applications for using hand posture and gesture recognition is sign language. A number of systems have been implemented that recognize various types of sign languages. For example, Starner was able to recognize a distinct forty word lexicon consisting of pronouns, nouns, verbs, and adjectives taken from American Sign Language with accuracies of over 90% [96]. A system developed by Kadous[48] recognized a 95 word lexicon from Australian Sign Language and achieved accuracies of approximately 80%. Murakami and Taguchi were able to adequately recognize 42 finger-alphabet symbols and 10 sign-language words taken from Japanese Sign Language[77]. Lu et al. [62] and Matuso[69] have also developed systems for recognizing a subset of Japanese Sign Language. Finally, Takahashi and Kishino were able to accurately recognize 34 hand gestures used in the Japanese Kana Manual Alphabet[105].

4.2 Gesture-to-Speech

Gesture-to-speech applications translate the user's hand postures and gestures into speech. Such systems could give hearing-impaired people the ability to communicate through a computer. A gesture-to-speech interface could be especially valuable to hearing-impaired people who wish to communicate with people who do not know sign language. Fels[35]and Kramer[53] have developed systems for converting hand gestures to speech that use gloves to collect the hand gestures and speech synthesizers for speech output.

4.3 Presentations

Baudel and Beaudouin-Lafon have developed an application that uses hand postures and gestures for giving presentations[9]. Their system, called Charade, gives the user the ability to control a computer-aided presentation. Charade uses the concept of a stationary "active zone", the zone in which gestures can be recognized; any gestures made outside the "active zone" are ignored. This lets the presenter make other gesture-based communications without having to worry that the system will recognize these gestures and create inappropriate commands. Charade gives the presenter gestural control of moving to the next or previous slide, marking a particular page, highlighting an area on the screen, and so on. Usability testing showed novice users achieved recognition accuracies of 72- to 84% while trained users obtained 90- to 98%.

4.4 Virtual Environments

Virtual environments (VE) are a common testbed and application domain for hand postures and gestures since the hands are the primary communication medium in VEs. One of the most important tasks a user must perform in many VE applications (e.g. architectural walkthrough, information visualization) is navigating through the VE. Among the many techniques for VE navigation (most of which are beyond the scope of this survey) is to use hand gestures for flying through the VE[73]. Typically, the user points in the direction he or she wants to go, and perhaps uses the other hand to control velocity. Object interaction is also necessary in a VE, and using hand gestures to interact with objects has attracted extensive research. The most traditional methods for interacting with objects is pointing, reaching and grabbing. Sturman, Zeltzer and Pieper[102], Davis[25], and Bryson[18] have all used hand gestures for object interaction in VEs. Rehag and Kanade have also developed hand posture and gesture recognition methods to create a 3D mouse for use in a virtual environment[91].

4.5 3D Modeling

3D modeling requires the user to create, manipulate and view 3D objects, and creation of these objects requires the user to specify a particular shape with the hands. For example, Krueger's VIDEODESK system allows the user to create 2D and 3D objects using the silhouette of the hand and pointing[55]. Weimer and Ganapathy use hand gestures to create B-spline-based 3D models[115], and Utsumi also uses static hand postures to create simple 3D primitives[108]. Manipulating already created models by translation, rotation and scale operations is also an important part of the modeling process, and Mapes presents a series of techniques for doing this type of object manipulation[66]. Using 3D hand gestures for modeling is relatively new and more research is required so that users can take advantage of their natural everyday movements for creating 3D models.

4.6 Multimodal Interaction

Another area in which hand postures and gestures are playing a significant role is multimodal interfaces. In many applications, hand posture and gesture recognition is incorporated with speech to create a more natural interface. In this type of multimodal integration the two modes can complement each other and make up for each other's recognition mistakes[83]. Although multimodal interaction has been around since the early 1980s[13], the paradigm is still in its infancy and is currently an active area of interface research. Lucente[63] has designed a multimodal interface that incorporates a vision-based hand posture and gesture recognition solution with speech input for a number of different applications (e.g. vacation planning, real-estate purchase). Multimodal interaction has also been applied to areas such as scientific visualization (see Figure 4.1) and room layout[58].

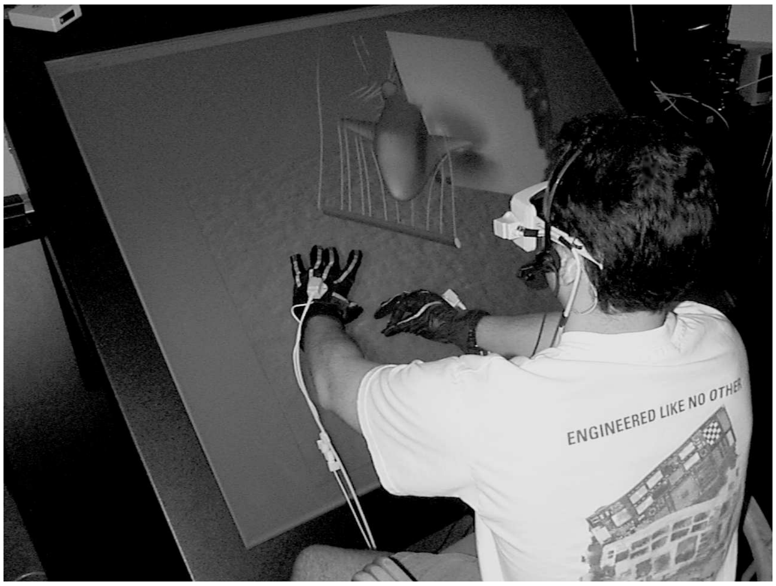

Figure 4.1: A user interacting with a dataset for visualizing a flow field around a space shuttle. The user manipulates the streamlines with his left hand and the shuttle with his right hand while viewing the data in stereo[58].

4.7 Human/Robot Manipulation and Instruction

An interesting application that can exploit hand postures and gestures is robot telemanipulation. An example of this type of telemanipulation was developed by Papper and Gigante to control a robot arm[84]. Hand postures and gestures can also be used to teach robots various commands and interactions by demonstration of the appropriate gestures by humans. Lee and Xu developed a system for teaching robots that can interactively learn new gestures after only a few training examples[60]. Tung and Kak have also developed a system for automatic learning of robot tasks using hand gestures[107].

4.8 Television Control

Another application for hand postures and gestures is control of audio and video devices. Freeman and Weissman have developed a system to control a television set by hand gestures[37]. Using an open hand, the user can turn the television on and off, change the channel, increase and decrease the volume, and mute the sound. Other applications that could use hand gestures are control of a VCR, a stereo, or a whole room[51].

5 Conclusions

Hand postures and gestures are an interesting interaction paradigm in a variety of computer applications. Two principal questions must be answered when using them. The first question is what technology to use for collecting raw data from the hand. Generally, two types of technologies are available for collecting this raw data. The first one is a glove input device, which measures a number of joint angles in the hand (except for the Pinch Gloves). Flexion and extension of the finger are the most common joints measured, although abduction, adduction, radial abduction, and palmer abduction can be also measured. Accuracy of a glove input device depends on the type of bend sensor technology used; usually, the more accurate the glove is, the more expensive it is. In many glove-based applications position and orientation data of the hand or hands must be collected by some tracking system, and many different technologies are used for this, including magnetic, inertial, and ultrasonic systems. The second way of collecting raw data is to use computer vision. In a vision-based solution, one or more cameras placed in the environment record hand movement. Both types of solutions have many advantages and disadvantages, and the question of which solution to use is a difficult one. However, when using a hand posture or gesture-based interface, the user does not want to wear the device and be physically attached to the computer. If vision-based solutions can overcome some of their difficulties and disadvantages, they appear to be the best choice for raw data collection.

The second question to be answered when using hand posture and gestures is what recognition technique will maximize accuracy and robustness. A number of recognition techniques are available and in some cases, the answer to the first question will narrow down the possibilities, since some of the recognition techniques work only for vision-based solutions. This survey has categorized these techniques into three broad categories:

- Feature extraction, statistics, and models
- Learning algorithms
- Miscellaneous techniques

Many of these techniques can be considered proven, but some of them have been reported in the literature only once, and this gives little indication that they are viable. The researcher can choose whether to use an established technique or one that requires more study. In addition, a recognition algorithm should be chosen on the basis of how many postures or gestures are in the recognition set, the complexity of the set, and whether or not the set is known beforehand.

There are a number of interesting areas for future research in hand posture and gesture recognition. The field is by no means mature – we have a long way to go before this type of metaphor is robust enough to be seen in commercial, mainstream applications. Research into better hardware for data collection is important. Better joint angle bend sensors, tracking systems, and faster processors will benefit the field immensely. Another possible area of research is to develop a glove input device that combines the qualities of the Pinch Glove and the CyberGlove and thus produce a more robust interface device[59]. There is a signifigant amount of research to be done in quantifying the validity of many of the techniques that have been reported only minimally in the literature, since it is unclear as to whether these techniques are viable without further analysis. Other areas of research include new techniques that allow robust and accurate recognition and fine-tuning established techniques to support larger and more complex posture and gesture sets.

This survey has provided a comprehensive overview of hand posture and gesture recognition techniques and technology. It gives researchers interested in starting work in this field an introduction to the various issues to be addressed when dealing with this type of interaction and allows them to have one document that references most of the viable literature. It also is useful as reference guide to this interaction paradigm.

Acknowledgments

Special thanks go to Timothy Miller, Robert Zeleznik, and Katrina Avery for valuable comments and discussions, and to Andries van Dam for support and encouragement. This work is supported in part by the NSF Graphics and Visualization Center, Advanced Networks and Services, Alias/Wavefront, International Business Machines, Microsoft, Sun Microsystems, and TACO.

Appendix A

Anatomy of the Human Hand

This section gives a brief introduction into the anatomical characteristics of the human hand. For a more comprehensive discussion of human hand anatomy, see Napier[78], the American Society for Surgery of the Hand[4], or the American Academy of Orthopedic Surgeons[3].

A.1 Hand and Finger Joints

The human hand is comprised of 17 joints that provide a total of 23 degrees of freedom (see Figure A.1). The fingers, labeled 2-5 in in Figure A.1, have three joints each. In order from fingertip to finger base, these joints are the distal interphalangeal (DIP), the proximal interphalangeal (PIP), and the metacarpophalangeal (MCP) . The thumb also has three joints, the thumb interphalangeal (Thumb IP), the thumb metacarpophalangeal (Thumb MP), and the trapeziometacarpal, in order from thumb tip to thumb base. The last two joints that make up the hand are the metacarpocarpal joints, located between the metacarpal and carpal bones on digits four and five.

A.2 Hand Motion

There are 23 degrees of freedom available in the hand above the wrist; if we include components of three dimensional movement, the degrees of freedom increase to 29. Figure A.2 shows a classification of hand motions. The DIP, PIP, Thumb IP, and Thumb MP joints in the four fingers are characterized by flexion and extension (a one degree of freedom movement). The MCPs in the four fingers are characterized by flexion, extension, hyperextension, and abduction (the separation or degree of bend between the fingers). The most complex movements the hand can perform are in the province of the trapeziometacarpal joint in the thumb: this joint's motion is characterized by radial and palmer abduction and anteposition, which is further classified by opposition and circumduction, and retroposition (see Figure A.2).

A.3 Muscles and Tendons in the Hand

Figure A.3 shows the muscles and tendons of the hand. There are many complexities and interconnections involved in the hand anatomy that contribute to the complexity of hand motion. An important group of tendons, connected to the extensor digitorum communis, are essentially the conduit between the extensor digitorum muscle and finger movements. This can have an adverse effect on measurement data from glove-based input devices[101].

A.4 Importance of the Hand's Anatomy

UI developers need a sound understanding of the human hand's anatomical structure in order to help them to determine what kinds of postures and gestures are easy and comfortable to make. For example, any posture or gesture requiring hyperextension of any of the fingers is not suitable since it puts more strain on the joints and tendons than the hand is accustomed to and thus can result in strain or injury. It is often useful to experiment with any postures or gestures than are going to be a part of an interface before they are implemented to make sure that they are comfortable, and do not put any

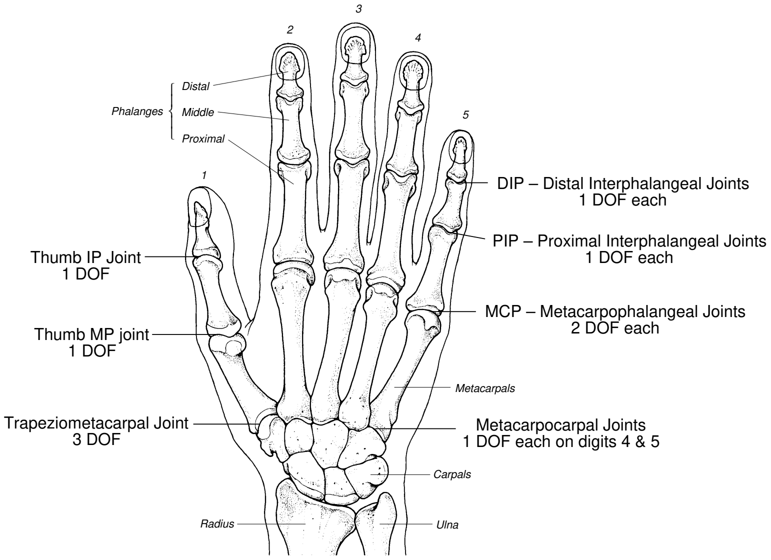

Figure A.1: The 17 joints in the hand and the associated 23 degrees of freedom (from Sturman[101]).

excessive strain on the hands.

A.5 Hand Models Used in Posture and Gesture Recognition

It is useful in recognizing hand postures and gestures, to have a model of the hand that can be used in recognition algorithms. The most common model is the angle-based hand model, which describes the hand using the joints angles and their associated degrees of freedom (see Figure A.1). The full model has a total of 29 data values, 23 associated with the degrees of freedom in the hand and six associated with its position and orientation. A major advantage of this hand model is that most glove-based input devices measure joint angles, so that no complex mathematical conversion is needed to move between glove data and model.

Another less common hand model is the point-based model, which has six tracked points, the center of the hand and the tips of each of the five digits[103]. Su[104] gives a mathematical derivation and conversion between the point-based hand model and the angle-based hand model. The advantage of the point-based hand model is that it is simpler to use in hand posture recognition since only six data points are used to classify the postures. It is especially suited to recognizing pointing, grabbing, and shooting postures used commonly in virtual environments, and also shows promise in recognizing American Sign Language.

However, the point-based hand model has distinct disadvantages. First, it restricts the hand posture and gesture space. The matrix computations for converting an angle-based hand model to a point-based model are complex and thus can increase response time. Also, error accumulation from the matrix computations can lead to large inaccuracies in fingertip positions. Ideally, one would use a point-based hand model directly, eliminating the conversion from an angle-based model. An area for future research is thus to develop a data glove that tracks the position and orientation of the fingertips instead of measuring joint angles, since no current input device to does this.

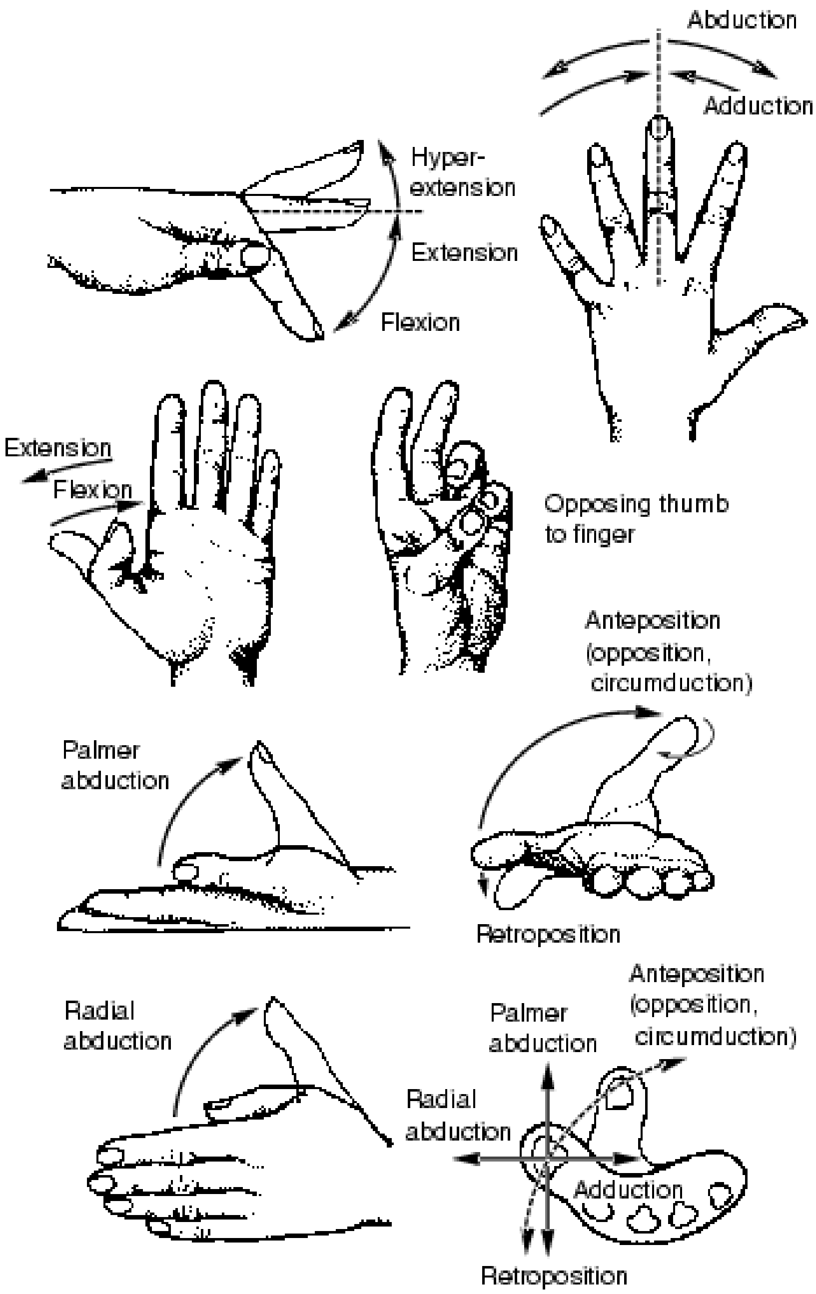

Figure A.2: The various motions that the hand and fingers can make using its 23 degrees of freedom (from Sturman[101]).

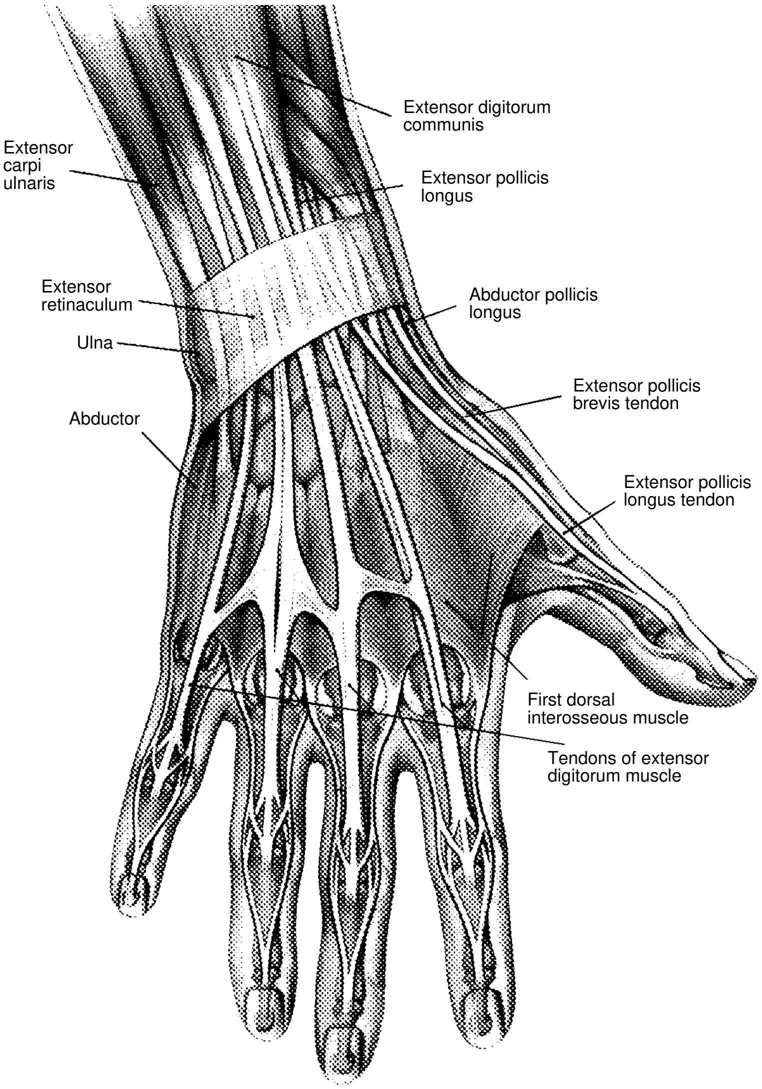

Figure A.3: The various muscles and tendons of the hand and wrist (from Kadous[48]).

Appendix B

Hand Posture and Gesture Classification

Hand posture and gesture help to augment spoken communication and also provide communication without speech. With an understanding and classification of how humans communicate with their hands, researchers can use hand postures and gestures more effectively as an interface to computer applications. Many hand posture and gesture classification systems and taxonomies have been developed; [29][49][72]; this appendix describes Sturman's Whole Hand Input Taxonomy[100], Nespoulous and Lecours' Gesture taxonomy[80], and the MIT Advanced Human Interface Group's (AHIG) Gesture Classification System[117].

B.1 Sturman's Whole Hand Input Taxonomy

Sturman's Whole Hand Input Taxonomy[100] is designed as a mapping between categories of hand actions and their interpretations. According to Sturman, "Hand actions are defined as position, motion, and forces generated by the hand. The interpretation of hand actions are the functional interpretation made by the user and/or the applications of the hand actions." Hand actions fall into two categories: continuous features and discrete features. Continuous features are based on the degrees of freedom of the hand and include such continuous quantities as fingertip position, joint velocities, and direction of motion. Hand gestures fall into this category, as do the forces on the pads of the fingers and palm. Discrete features are based on static values of the features of the hand. Hand postures, such as a fist or a pointing posture, fall into the discrete feature category.

Interpretation of hand actions is divided into three categories: direct, mapped, and symbolic interpretation. In direct interpretation, the user is physically interacting with the virtual world as if it were the real world; as when users grab a virtual world object and place it on a virtual table in the same way they would grab a real coffee mug and place it on a real table. Direct interpretation also includes interaction in which the hand mimics the actions of the object being controlled. In a mapped interpretation, data from the hand is mapped to some virtual input device such as a button or slider; as when the flexion of the index finger to manipulates a slider that changes an interocular distance parameter for stereoscopic viewing. Finally, in symbolic interpretation, users specify a hand posture or gesture that is cognitively mapped to some function or task. For example, a series of hand gestures can signify a token stream used in the recognition of American Sign Language (ASL) or movement through a virtual environment.

Using the two categories of hand action and the three categories of interpretation, Sturman derives six categories that classify whole-hand input:

Continuous/Direct: Continuous hand data is mapped to a kinematically similar action: a graphical hand follows a user's real hand motion.

Continuous/Mapped: Continuous hand data is mapped to some logical input device: a finger is used to position a mouse cursor.

Continuous/Symbolic: The application interprets continuous hand data and maps it to a particular task: in flying through a virtual environment, the distance between the two hands determines speed and the vector made by the two hands determines direction[73].

Discrete/Direct: Discrete hand data or a hand posture is mapped to a directly manipulative task. Sturman claims that this category is rarely used because it has few applications.

Discrete/Mapped: Discrete hand data is mapped to a discrete activation level: an object is animated as long as the user makes a fist.

Discrete/Symbolic: Discrete hand data is used to generate commands in an application: a user makes a halt posture to make an object stop moving.

B.2 Nespoulous and Lecours' Gesture Taxonomy

Nespoulous and Lecours' taxonomy[80] defines and groups different types of gestures in terms of movement and interpretation (see Figure B.1). This taxonomy is also discussed in Quek[87]. Gestures are divided into two categories: acts and symbols. Act gestures are movements that relate directly to the intended interpretation.

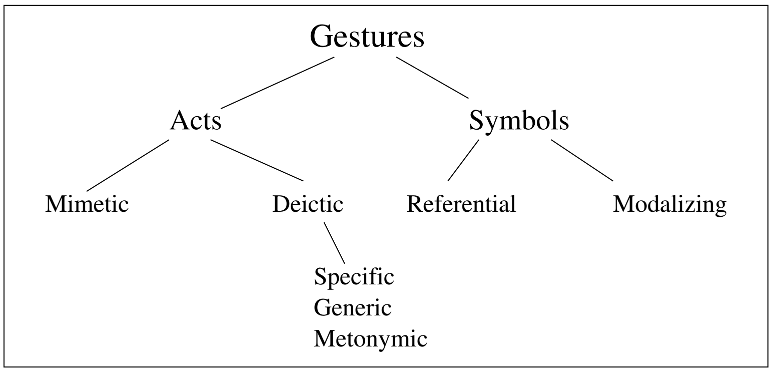

Figure B.1: Taxonomy of gestures described by Quek (adapted from Quek[87]).

Symbolic gestures are based on movements that require some level of knowledge to interpret. Symbolic gestures are classified as referential and modalizing. Referential gestures are gestures that refer to some object or concept; for example, putting a gun posture up to one's head often indicates that one has made a foolish mistake. Modalizing gestures are gestures used in conjunction with another input modality (e.g. speech).

For example, in asking whether someone had seen a particular person, extending the hand out at a certain level could indicate the person's height.

Act gestures are broken up into mimetic and deictic gestures. Mimetic gestures are those that mimic or pantomime some task to be performed, deictic gestures are gestures that derive from pointing. They can be further broken up into specific, generic, and metonymic gestures. All three forms are physically performed in the same fashion but they are interpreted differently based on context. Specific deictic gestures are made when the user selects a particular object or location. Generic deictic gestures are used when pointing to a member of a group of objects with the intention of discussing the group itself. Metonymic deictic gestures are made when pointing to an object to signify an entity related to it.

B.3 MIT AHIG's Gesture Classification System

The AHIG gesture classification system was first discussed in Wexelblat[117], is also indirectly discussed in Cassell[20] and Wilson et al.[118]. AHIG's classification system starts from the idea that previous gesture classification systems, such as Efron's[29] and Kendon's[49], are oriented to the psychological domain and do not necessarily apply to computer applications. The system has broken five major categories:

- Symbolic/modalizing
- Pantomimic
- Iconic
- Deictic/lakoff
- Beat/Butterworth's/Self-adjusters

Symbolic gestures are essentially hand postures used to represent an object or concept, and are always directly mapped to a particular meaning: for instance, the 'thumbs up' posture means that everything is okay. Modalizing gestures (see Appendix B.2 above) are also included in this category.

Pantomimic gestures involve using the hands to represent a task or interaction with a physical object. Users making this type of gesture mimic an action they would do if they were actually interacting in the real world: for example, making a swinging gesture with one's hands to indicate hitting a baseball with a bat.

Iconic gestures are gestures that represent an object. The hands become the object or objects discussed. These gestures usually are performed to act out a particular event in which the representative object is the focal point such as someone pretending to drive a car.

Deictic gesture or pointing gestures are used to indicate a particular object, as discussed in appendix B.2 above. The other type of gesture included in this category are Lakoff gestures[57], associated verbal utterances that specify a particular metaphor such as happiness or anger. A gesture usually accompanies these utterances to show the directionality of the metaphor.

The last category contains three types of gestures: beats, Butterworth's, and self-adjusters. Beats are gestures used for emphasis, especially when used with speech. Beat gestures can help speakers emphasize particular words or concepts and also help direct the listener's attention. Butterworth gestures[19] are similar to beats except they are primarily used to mark unimportant events. The classic example of a Butterworth gesture is 'hand waving' as a placeholder for speaking when one is still thinking about how to say something. Finally, self-adjusters are gestures people make when they fidget: for example, when one taps a finger or moves a foot rapidly.

Bibliography

[1] Agarwal, R., and J. Sklansky. Estimating Optical Flow from Clustered Trajectories in Velocity-Time. In *Proceedings of the 11th IAPR International Conference on Pattern Recognition*. Vol. #1. Conference A: Computer Vision and Applications, 215-219, 1992.

[2] Aha, David W., Dennis Kibler, and Marc K. Albert. Instance-based Learning Algorithms. *Machine Learning*, 6, 37-66, 1991.

[3] American Academy of Orthopedic Surgeons. *Joint Motion: Method of Measuring and Recording*. Churchill Livingstone, New York, 1988.

[4] American Society for Surgery of the Hand. *The Hand: Examination and Diagnosis*. Churchill Livingstone, New York, 1978.

[5] Anderson, James A. *An Introduction to Neural Networks*. Bradford Books, Boston, 1995.

[6] Ascension Technology Corporation. The Flock of Birds Installation and Operation Guide. Burlington, Vermont, 1996.

[7] Auer, T., A. Pinz, and M. Gervautz. Tracking in a Multi-User Augmented Reality System. In *Proceedings of the IASTED International Conference on Computer Graphics and Imaging*, 249-253, 1998.

[8] Banarse, D. S. Hand Posture Recognition with the Neocognitron Network. School of Electronic Engineering and Computer Systems, University College of North Wales, Bangor, 1993.

[9] Baudel, Thomas, and Michel Beaudouin-Lafon. Charade: Remote Control of Objects Using Free-Hand Gestures. *Communications of the ACM*, 36(7):28-35, 1993.

[10] Blake, Andrew, and Michael Isard. 3D Position, Attitude and Shape Input Using Video Tracking of Hands and Lips. In *Proceedings of SIGGRAPH'94*, ACM Press, 185-192, 1994.

[11] Birk, Henrik, Thomas B. Moeslund, and Claus B. Madsen. Real-Time Recognition of Hand Alphabet Gestures Using Principal Component Analysis. In *Proceedings of The 10th Scandinavian Conference on Image Analysis*, 1997.

[12] Birk, Henrik and Thomas B. Moeslund. Recognizing Gestures from the Hand Alphabet Using Principal Component Analysis. Master's thesis, Aalborg University, 1996.

[13] Bolt, R.A. Put That There: Voice and Gesture at the Graphics Interface. In *Proceedings of SIGGRAPH'80*, ACM Press, 262-270, 1980.

[14] Brand, Matthew, and Irfan Essa. Causal Analysis for Visual Gesture Understanding. MIT Media Laboratory Perceptual Computing Section Technical Report No. 327, 1995.

[15] Brand, Matthew. Explanation-Mediated Vision: Making Sense of the World with Causal Analysis. Ph.D dissertation, Northwestern Univeristy, 1994.

[16] Brand, Matthew, Lawrence Birnbaum, and Paul Cooper. Sensible Scenes: Visual Understanding of Complex Structures Through Causal Analysis. In *Proceedings of the 1993 AAAI Conference*, 49-56, 1993.

[17] Broomhead, D., and D. Lowe. Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems*, 2:321-355, 1988.

[18] Bryson, Steve. Virtual Reality in Scientific Visualization. *Communications of the ACM*, 39(5):62-71, 1996.

[19] Butterworth, B., and G. Beattie. Gesture and Silence as Indicators of Planning in Speech. In *Recent Advances in the Psychology of Language*, Campbell and Smith (eds.), Plenum Press, New York, 1978.

[20] Cassell, Justine. A Framework for Gesture Generation and Interpretation. In *Computer Vision in Human-Machine Interaction*. R. Cipolla and A. Pentland (eds.), Cambridge University Press, forthcoming.

[21] Charniak, Eugene. *Statistical Language Learning*. MIT Press, Cambridge, 1993.

[22] Cootes, T.F., C.J. Taylor, D.H. Cooper, and J. Graham. Active Shape Models – Their Training and Applications. *Computer Vision and Image Understanding* , 61(2), January, 1995.

[23] Cootes, T.F., and C.J. Taylor. Active Shape Models – 'Smart Snakes'. In *Proceedings of the British Machine Vision Conference*, Springer-Verlag, 266-275, 1992.

[24] Davis, James William. Appearance-Based Motion Recognition of Human Actions. Master's thesis, Massachusetts Institute of Technology, 1996.

[25] Davis, James, and Mubarak Shah. Gesture Recognition. Technical Report, Department of Computer Science, University of Central Florida, CS-TR-93-11, 1993.

[26] Defanti, Thomas, and Daniel Sandin. Final Report to the National Endowment of the Arts. US NEA R60-34-163, University of Illinois at Chicago Circle, Chicago, Illinois, 1977.

[27] Darrell, Trevor J., and Alex P. Pentland. Recognition of Space-Time Gestures Using a Distributed Representation. MIT Media Laboratory Vision and Modeling Group Technical Report No. 197, 1993.

[28] Dorner, Brigitte. Chasing the Colour Glove: Visual Hand Tracking. Master's thesis, Simon Fraser University, 1994.

[29] Efron, D. *Gesture and Environments.* King's Crown Press, Morningside Heights, New York, 1941.

[30] Eglowstein, Howard. Reach Out and Touch Your Data. *Byte*, July, 283-290, 1990.

[31] Encarnação, M. A Survey on Input Technology for the Virtual Table Interface Device. Technical Report, Fraunhofer Center for Research in Computer Graphics, Inc, 1997.

[32] Everitt, B.S. *Cluster Analysis.* John Wiley and Sons, New York, 1974.

[33] Fakespace. Pinch Glove System Installation Guide and User Handbook, Mountain View, California, 1997.

[34] Fels, Sidney, and Geoffrey Hinton. Glove-TalkII: An Adaptive Gesture-to-Format Interface. In *Proceedings of CHI'95 Human Factors in Computing Systems*, ACM Press, 456-463, 1995.

[35] Fels, Sidney. Glove-TalkII: Mapping Hand Gestures to Speech Using Neural Networks – An Approach to Building Adaptive Interfaces. Ph.D. dissertation, University of Toronto, 1994.

[36] Fifth Dimension Technologies. http://www.5dt.com/products.html, 1999.

[37] Freeman, William T., and Craig D. Weissman. Television Control by Hand Gestures. Technical Report, Mitsubishi Electronic Research Laboratories, TR-94-24, 1994.

[38] Fukushima, Kunihiko. Analysis of the Process of Visual Pattern Recognition by the Neocognitron. *Neural Networks*, 2:413-420, 1989.

[39] General Reality Company. GloveGRASP User's Guide. San Jose, California, 1996.

[40] Glassner, Andrew. *Principles of Digital Image Synthesis.* Morgan Kaufman, San Francisco, 1995.

[41] Grimes, G. Digital Data Entry Glove Interface Device. Bell Telephone Laboratories, Murray Hill, New Jersey. US Patent Number 4,414,537, 1983.

[42] Grimson, W.E. *Object Recognition by Computer: The Role of Geometric Constrants.* MIT Press, Boston, 1990.

[43] Hand, Chris, Ian Sexton, and Michael Mullan. A Linguistic Approach to the Recognition of Hand Gestures. In *Proceedings of the Designing Future Interaction Conference*, University of Warwick, UK, 1994.

[44] Heap, A. J., and F. Samaria. Real-Time Hand Tracking and Gesture Recognition Using Smart Snakes. In *Proceedings of Interface to Real and Virtual Worlds*, Montpellier, June 1995.

[45] Huang, X. D., Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition.* Edinburgh University Press, Edinburgh, 1990.

[46] InterSense. http://www.isense.com, 1999.

[47] Jolliffe, I. T. *Principal Component Analysis.* Springer-Verlag, New York, 1986.

[48] Kadous, Waleed. GRASP: Recognition of Australian Sign Language Using Instrumented Gloves. Bachelor's thesis, University of New South Wales, 1995.

[49] Kendon, Adam. Current Issues in the Study of Gesture. In *The Biological Foundations of Gestures: Motor and Semiotic Aspects.* Nespoulous, Perron, and Lecours (eds.), Lawrence Erlbaum Associates, Hillsday, NJ, 1986.

[50] Kessler, G. Drew, Larry H. Hodges, and Neff Walker. Evaluation of the CyberGlove as a Whole Hand Input Device. *ACM Transactions on Computer-Human Interaction*, 2(4):263-283, 1995.

[51] Kohler, Marcus. Special Topics of Gesture Recognition Applied to Intelligent Home Environments. In *Proceedings of the International Gesture Workshop'97*, Berlin, 285-297, 1997.

[52] Kramer, James. Communication System for Deaf, Deaf-blind, and Non-vocal Individuals Using Instrumented Gloves. Virtual Technologies, US Patent Number 5,047,952, 1991.

[53] Kramer, James, and Larry Leifer. The Talking Glove: An Expressive and Receptive 'Verbal' Communication Aid for the Deaf, Deaf-blind, and Non-vocal. Technical Report, Department of Electrical Engineering, Stanford University, 1989.

[54] Krose, Ben J. A., and P. Patrick van der Smagt. *An Introduction to Neural Networks.* University of Amsterdam, 1995.

[55] Krueger, Myron W. *Artificial Reality II.* Addison-Wesley Publishing Company, New York, 1991.

[56] Kumo, Yoshinori, Tomoyuki Ishiyama, Kang-Hyun Jo, Nobutaka Shimada and Yoshiaki Shirai. Vision-Based Human Interface System: Selectively Recognizing Intentional Hand Gestures. In *Proceedings of the IASTED International Conference on Computer Graphics and Imaging*, 219-223, 1998.

[57] Lakoff, G., and M. Johnson. *Metaphors We Live By.* University of Chicago Press, Chicago, 1980.

[58] LaViola, Joseph. A Multimodal Interface Framework For Using Hand Gestures and Speech in Virtual Environment Applications. To appear in *Proceedings of the Gesture Workshop'99*, Gif-sur-Yvette, 1999.

[59] LaViola, J., and R. Zeleznik. Flex and Pinch: A Case Study of Whole Hand Input Design for Virtual Environment Interaction. Submitted to *IASTED International Conference on Computer Graphics and Imaging*, 1999.

[60] Lee, Christopher, and Yangsheng Xu. Online Interactive Learning of Gestures for Human/Robot Interfaces. In *1996 IEEE International Conference on Robotics and Automation*, vol. 4, 2982-2987, 1996.

[61] Liang, Rung-Huei, and Ming Ouhyoung. A Sign Language Recognition System Using Hidden Markov Model and Context Sensitive Search. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology'96*, ACM Press, 59-66, 1996.

[62] Lu, Shan, Seiji Igi, Hideaki Matsuo, and Yuji Nagashima. Towards a Dialogue System Based on Recognition and Synthesis of Japanese Sign Language. In *Proceedings of the International Gesture Workshop'97*, Berlin, 259-272, 1997.

[63] Lucente, Mark, Gert-Jan Zwart, and Andrew D. George. Visualization Space: A Testbed for Deviceless Multimodal User Interface. In *Intelligent Environments 98*, AAAI Spring Symposium Series, 87-92, 1998.

[64] Makower, J., M. Parnianpour, and M. Nordin. The Validity Assessment of the Dextrous Hand Master: A Linkage System for the Measurement of the Joints of the Hand. In *Abstracts of the First World Congress of Biomechanics.* (Volume #2), La Jolla, California, 338, 1990.

[65] Mann, Steve. Wearable Computing: A First Step Toward Personal Imaging, *IEEE Computer*, 30(2): 25-32, 1997.

[66] Mapes, Daniel J., and Michael J. Moshell. A Two-Handed Interface for Object Manipulation in Virtual Environments. In *PRESENSE: Teleoperators and Virtual Environments*, 4(4):403-416, 1995.

[67] Marcus, Beth A., and Philip J. Churchill. Sensing Human Hand Motions for Controlling Dexterous Robots. In *The Second Annual Space Operations Automation and Robotics Workshop*, Wright State University, June 20-23, 1988.

[68] Martin, Jerome, and James L. Crowley. An Appearance-Based Approach to Gesture Recognition. In *Proceedings of the Ninth International Conference on Image Analysis and Processing*, 340-347, 1997.

[69] Matsuo, Hideaki, Seiji Igi, Shan Lu, Yuji Nagashima, Yuji Takata, and Terutaka Teshima. The Recognition Algorithm with Non-contact for Japanese Sign Lan-

guage Using Morphological Analysis. In *Proceedings of the International Gesture Workshop'97* , Berlin, 273-284, 1997.

[70] Maybeck, Peter S. *Stochastic Models, Estimation and Control.* Volume 1, Academic Press, Inc., 1979.

[71] Mehrotra, Kishan, Chilukuri K. Mohan, and Sanjay Ranka. *Elements of Artifical Neural Networks.* The MIT Press, Boston, 1997.

[72] McNeill, D. and E. Levy. Conceptual Representations in Language Activity and Gesture. In *Speech, Place, and Action*, Jarvella and Klein (eds.), John Wiley and Sons Ltd., New York, 1982.

[73] Mine, Mark. Moving Objects In Space: Exploiting Proprioception In Virtual Environment Interaction. In *Proceedings of SIGGRAPH'97*, ACM Press, 19-26, 1997.

[74] Mitchell, Tom M. *Machine Learning.* McGraw-Hill, Boston, 1997.

[75] Mulder, Axel. Human Movement, Tracking Technology. Technical Report, School of Kinesiology, Simon Fraser University, 94-1, 1994.

[76] Multigen. SmartScene Video Clip, Discovery Channel's NextStep program, 1998.

[77] Murakami, Kouichi, and Hitomi Taguchi. Gesture Recognition Using Recurrent Neural Networks. In *Proceedings of CHI'91 Human Factors in Computing Systems*, 237-242, 1991.

[78] Napier, John. *Hands*, Pantheon Books, New York, 1980.

[79] Nam, Yanghee, and KwangYun Wohn. Recognition of Space-Time Hand-Gestures Using Hidden Markov Model. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology'96*, ACM Press, 51-58, 1996.

[80] Nespoulous, J., and A. R. Lecours. Gestures, Nature and Function. In *The Biological Foundations of Gestures: Motor and Semiotic Aspects.* Nespoulous, Perron, and Lecours (eds.), Lawrence Erlbaum Associates, Hillsday, NJ, 1986.

[81] Newby, Gregory B. Gesture Recognition Using Statistical Similarity. In *Proceedings of Virtual Reality and Persons with Disabilities*, 1993.

[82] Nissho Electronics Corporation. Introduction to SuperGlove. Tokyo, Japan, 1997.

[83] Oviatt, Sharon, and Robert VanGent. Error Resolution During Multimodal Human-Computer Interaction. In *Proceedings of the International Conference on Spoken Language Processing*, 204-207, 1996.

[84] Papper, M., and M. Gigante. Using Gestures to Control a Virtual Arm. In *Virtual Reality Systems*, R. Earnshaw, H. Jones, and M. Gigante (eds.), Academic Press, London, 1993.

[85] Pausch, Randy. Virtual Reality on Five Dollars a Day. Technical Report CS-91-21, Department of Computer Science, University of Virginia, 1991.

[86] Polhemus. http://www.polhemus.com, 1999.

[87] Quek, Francis K.H. Toward a Vision-Based Gesture Interface. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology'94*, ACM Press, 17-31, 1994.

[88] Rabiner, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* 77(2):267-296, 1989.

[89] Rabiner, L. R., and B.H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 4-16, January, 1986.

[90] Rangarajan, K., and M. Shah. Establishing Motion Correspondence. *CVGIP: Image Understanding*, 54:56-73, 1991.

[91] Rehag, James M., and Takeo Kanade. DigitEyes: Vision-Based Human Hand Tracking. Technical Report CMU-CS-93-220, School of Computer Science, Carnegie Mellon University, 1993.

[92] Rubine, Dean. Specifing Gestures by Example. In *Proceedings of SIGGRAPH'91*, ACM Press, 329-337, 1991.

[93] Russell, Stuart, and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[94] Schlenzig, Jennifer, Edward Hunter, and Ramesh Jain. Recursive Spatio-Temporal Analysis: Understanding Gestures. Technical Report VCL-95-109, Visual Computing Laboratory, University of California, San Diego, 1995.

[95] Sirovich, I., and M. Kirby. Low-dimensional Procedure for the Characterization of Human Faces. *Journal of the Optical Society of America*, 4(3):519-524, 1987.

[96] Starner, Thad, and Alex Pentland. Real-Time American Sign Language Recognition from Video Using Hidden Markov Models. MIT Media Laboratory Perceptual Computing Section Technical Report No. 375, 1996.

[97] Starner, Thad. Visual Recognition of American Sign Language Using Hidden Markov Models. Master's thesis, Massachusetts Institute of Technology, 1995.

[98] State, Andrei, Hirota Gentaro, David T. Chen, William F. Garrett, and Mark A. Livingston. Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. In *Proceedings of SIGGRAPH'96*, ACM Press, 429-438, 1996.

[99] Sturman, David J., and David Zeltzer. A Survey of Glove-based Input. *IEEE Computer Graphics and Applications*, 14(1):30-39, 1994.

[100] Sturman, David J., and David Zeltzer. A Design Method for 'Whole-Hand' Human-Computer Interaction. *ACM Transactions on Information Systems*, 11(3):219-238, 1993.

[101] Sturman, David J. Whole-hand Input. Ph.D dissertation, Massachusetts Institute of Technology, 1992.

[102] Sturman, David J., David Zeltzer, and Steve Pieper. Hands-on Interaction with Virtual Environments. In *Proceedings of the ACM SIGGRAPH Symposium on User Interface Software and Technology'89*, ACM Press, 19-24, 1989.

[103] Su, S. Augustine, and Richard Furuta. A Logical Hand Device in Virtual Environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ACM Press, 33-42, 1994.

[104] Su, S. Augustine. Hand Modeling in Virtual Environment. Master's Degree Scholarly Paper, Department of Computer Science, University of Maryland, College Park, 1993.

[105] Takahashi, Tomoichi, and Fumio Kishino. Hand Gesture Coding Based on Experiments Using a Hand Gesture Interface Device. *SIGCHI Bulletin* 23(2):67-73, 1991.

[106] Turk, M., and A. Pentland. Eigenfaces for Recognition. *Journal of Neuroscience*, 3(1):71-86, 1991.

[107] Tung, C. P., and A. C. Kak. Automatic Learning of Assembly Tasks Using a Dataglove System. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, 1-8, 1995.

[108] Utsumi, Akira, Jun Kurumisawa, Takahiro Otsuka, and Jun Ohya. Direct Manipulation Scene Creation in 3D. SIGGRAPH'97 Electronic Garden, 1997.

[109] Veltman, S. R., and R. Prasad. Hidden Markov Models Applied to On-line Handwritten Isolated Character Recognition. *IEEE Transactions on Image Processing*, 314-318, 1994.

[110] Virtual Technologies. http://www.virtex.com/prod_CyberGloveTM.html, 1999.

[111] Virtual Technologies. CyberGlove User's Manual. Palo Alto, California, 1993.

[112] Watson, Richard. A Survey of Gesture Recognition Techniques. Technical Report TCD-CS-93-11, Department of Computer Science, Trinity College Dublin, 1993.

[113] Welch, Greg, and Gary Bishop. SCAAT: Incremental Tracking with Incomplete Information. In *Proceedings of SIGGRAPH'97*, ACM Press, 333-345, 1997.

[114] Welch, Greg, and Gary Bishop. An Introduction to the Kalman Filter. Technical Report TR 05-041, Department of Computer Science, University of North Carolina at Chapel Hill, 1995.

[115] Weimer, D. and S. K. Ganapathy. Interaction Techniques Using Hand Tracking and Speech Recognition. In *Multimedia Interface Design*, Meera M. Blattner and Roger B. Dannenberg, (eds.), Addison-Wesley Publishing Company, New York, 109-126, 1992.

[116] Wexelblat, Alan. An Approach to Natural Gesture in Virtual Environments. *ACM Transactions on Computer-Human Interaction*, 2(3):179-200, 1995.

[117] Wexelblat, Alan. A Feature-Based Approach to Continuous-Gesture Analysis. Master's thesis, Massachusetts Institute of Technology, 1994.

[118] Wilson, Andrew D., Aaron F. Bobick and Justine Cassell. Recovering the Temporal Structure of Natural Gesture. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996.

[119] Wise, Sam, William Gardner, Eric Sabelman, Erik Valainis, Yuriko Wong, Karen Glass, John Drace, and Joseph Rosen. Evaluation of a Fiber Optic Glove for Semiautomated Goniometric Measurements. *Journal of Rehabilitation Research and Development*, 27(4): 411-424, 1990.

[120] Youngblut, C., R.E. Johnson, S.H. Nash, R.A. Wienclaw, and C.A. Will. Review of Virtual Environment Interface Technology. Technical Report IDA Paper P-3186, Log: H96-001239. Institute for Defense Analysis, 1996.

[121] Zimmerman, Thomas G., Jaron Lanier, Chuck Blanchard, Steve Bryson, and Young Harvill. A Hand Gesture Interface Device. In *Proceedings of CHI+GI'87 Human Factors in Computing Systems and Graphics Interface*, ACM Press, 189-192, 1987.

[122] Zimmerman, Thomas G. Optical Flex Sensor. US Patent Number 4,542,291, 1985.