A Survey of
Gesture Recognition Techniques
Technical Report TCD-CS-93-11

Richard Watson,
Department of Computer Science,
Trinity College,
Dublin 2.
Richard.Watson@cs.tcd.ie

17 July 1993

Abstract

Processing speeds have increased dramatically, bitmapped displays allow graphics to be rendered and updated at increasing rates, and in general computers have advanced to the point where they can assist humans in complex tasks. Yet input technologies seem to cause the major bottleneck in performing these tasks: under-utilising the available resources, and restricting the expressiveness of application use.

We use our hands constantly to interact with things: pick them up, move them, transform their shape, or activate them in some way. In the same unconscious way, we gesticulate in communicating fundamental ideas: 'stop', 'come closer', 'over there', 'no', 'agreed', and so on. Gestures are thus a natural and intuitive form of both interaction and communication.

This report develops the motivations for gestural input and surveys current gesture recognition techniques. A recognition technique under development at TCD, as part of the GLAD-IN-ART (EP5363) project, is also introduced.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Enabling Technologies 2

1.2.1 Instrumented Gloves 2

1.2.2 Vision-based Tracking 3

1.3 Gesture Based Applications 4

1.3.1 Physically-based Manipulation 4

1.3.2 Linguistic Gestures 6

1.4 Gesture Recognition at TCD. 8

2 Approaches to the Problem **11**

2.1 First Steps 11

2.2 Template Matching 12

2.2.1 VPL's Posture Recognition 12

2.2.2 Extending Posture templates to Gestures 13

2.2.3 Syntactic Classification 13

2.3 Neural Networks 15

2.3.1 Fels' GloveTalk 16

2.3.2 Brooks' Kohohen Net approach 17

2.3.3 Beale and Edwards' Postures 17

2.4 Statistical Classification 18

2.4.1 Rubine's stroke recognition 18

2.4.2 The features 19

2.4.3 Classification 20

2.4.4 Extending the System 20

2.5 Discontinuity Matching 21

2.6 The related problems of handwriting and speech recognition . . . 22

Chapter 1

Introduction

1.1 Motivation

Processing speeds have increased dramatically, bitmapped displays allow graphics to be rendered and updated at alarming rates, and in general computers have advanced to the point where they can assist humans in complex tasks. Yet input technologies seem to cause the major bottleneck in performing these tasks: under-utilising the available resources. This is regrettable, since according to Paul McAvinney, "most of the useful information in the world resides in humans, not computers". Users spend the majority of the time they interact with computers inputting information. Thus, the total time to perform any task would hardly improve if processing was to become infinitely fast.

We use our hands constantly to interact with things: pick them up, move them, transform their shape, or activate them in some way. In the same unconscious way, we gesticulate in communicating fundamental ideas: 'stop', 'come closer', 'over there', 'no', 'agreed', and so on. Gestures are thus a natural and intuitive form of both interaction and communication.

Recognising this, researchers are developing devices that allow gestures to be used as a form of input. As discussed below, some of these devices are commercially available, most popularly in the form of gloves worn on the hand.

One of the most important concepts in **H**uman **C**omputer **I**nteraction is that of direct manipulation. This refers to the user's experience of interaction being directly with the objects of interest rather than through an intermediary system. The Apple Macintosh Desktop uses direct manipulation for most of its operations. To move files from one folder to another, the user clicks the mouse on the files to be moved and drags them to the target folder. In a traditional shell, the command to move files is typed and the corresponding operation is performed, from the user's perspective by something else, the OS in this case.

Hutchins [16] refers to *directness* as an impression or feeling about an interface resulting from the commitment of fewer cognitive resources. In less abstract words, the more a user has to think about an interface, the more removed he is from the task. This distance is referred to as "the gulf between the user's goals and the way they must be specified".

There are two modes of interaction that are often not clearly delineated in the literature: gestures as a symbolic language and gesturing to provide multi-dimensional control. As interface paradigms, both move towards more natural[1] direct manipulation, aiming to render the computer transparent in using an application. In the survey of gesture-based applications below, the two types of approach are made explicit, where possible. Often, however, the two modes are mixed as in the VIEW system [37]—where objects are directly manipulated by the hands in 3-dimensions *and* system commands are issued using different gestures. The two types of interaction, "physically-based manipulation" and "linguistic gestures" are similarly viewed by Weimer and Gannapathy [43] as being at opposite ends of the direct manipulation spectrum.

1.2 Enabling Technologies

1.2.1 Instrumented Gloves

The **'Sayre' Glove** was the first instrumented glove to be invented, c. 1977, based on an idea from Rich Sayre, it used flexible tubes with a light source at one end and a photocell at the other. Finger flexion was thus measured by the amount of light incident on the photocell.

In 1983, Gary Grimes received a patent [14] covering the use of a special electronic glove solely to interpret a manual alphabet for data entry, for his **Digital Data Entry Glove**. The glove has specifically positioned flex sensors capable of recognising an 80 character superset of the Single Hand Manual Alphabet for the Deaf.

By far the most successful glove is the **VPL DataGlove**, developed by Zimmerman [45]. The DataGlove is based on patented optical fibre sensors along the backs of the fingers, two for each finger. Like the Sayre glove, finger flexion bends the fibres, attenuating the light they transmit. This analog signal is sent to a processor which determines the joint angles based on calibrations for each user. Posture recognition software is included with the DataGlove so that users can map configurations of joints to commands, as described in section 2.2.1.

Position and orientation of the hand is calculated using a Polhemus 3SPACE sensor, which tracks the 6 degrees of freedom using the principle of electro-

[1] Actions that are natural also tend to be ingrained or intuitive behaviour. Interface designers must be mindful of not conflicting with natural or trained responses, like the lesson learnt from the early USAF F-111 swing-wing aircraft in which the faster wing configuration was initiated by pulling the stick backwards—a motion that means "slower" to pilots—leading to several landing accidents before the controls were reversed.

magnetic induction. The tracker consists of a fixed transmitter and a small lightweight cubic receiver which is mounted on the glove, both contain three mutually perpendicular coils. The relationship between current flow in the transmitter's coils and the induced current in the receiver's coils is the basis for computing the position and orientation.

While the DataGlove emphasised user comfort with reasonable precision[2], the **Dextrous HandMaster** developed by Exos is a far more accurate, but less comfortable, glove device [27]. It consists of an intricate exoskeleton of aluminium that is fitted to the back of the hand. All twenty of the hand joints have Hall-effect sensors to measure the bending angle.

Inspired by the success of the VPL DataGlove, the toy manufacturers Mattel, released the **PowerGlove** in 1989, as a low cost and hard-wearing controller for Nintendo games consoles [9]. The PowerGlove uses resistive ink sensors embedded into flexible molded plastic on the back of the hand and fingers. The total flexion of each finger is measured by one resistive ink sensor—making the PowerGlove the least accurate of the three competing gloves.

The PowerGlove also provides 6D pose (position and orientation) information, based on ultrasonic waves. Two ultrasonic transmitters attached to opposite sides of the glove transmit ultrasonic clicks. Three receivers placed at the corners of a video monitor receive the click and based on *a priori* knowledge of the relative locations of the transmitters can calculate the triangulation.

Researchers and hobbyists in this field often opt for Mattel's PowerGlove over the other two because of the enormous gulf in price: Exos DHM,$15,000; VPL DataGlove, $8,000; Mattel PowerGlove, $20 !

1.2.2 Vision-based Tracking

Poizner and other researchers at the Salk Institute used a camera-based LED system to analyse sign language [18]. Analysis was off-line, avoiding computational problems. Various analytical techniques were proposed from which to qualify the linguistically significant features of signed language. Although it was not mentioned in the paper, the system could be used to track hand motion for gestures.

In his responsive environments, Krueger tracks participants using a single video camera [24]. Image analysis is simplified by using the silhouette image of body motion. Rheingold [15] describes Krueger's work in detail, and of the image processing in particular he gives the rationale for a simplification, "in the context of a human body, the highest line of a silhouette means the top of the head, the rightmost endpoint of a skinny edge means a fingertip".

O'Neill [30] reports on the development of a passive pose tracking system based on tracking icons with a stereo camera system. The application of com-

[2] Official finger flex accuracy is given at 1°, but formal and informal third-party analysis have shown actual accuracy to be nearer 5°

puter vision techniques removes the need for intrusive cabling and makes tracking multiple objects feasible.

1.3 Gesture Based Applications

1.3.1 Physically-based Manipulation

3D Design Applications such as CAD and telerobotics form the platform of applications that require 3D input to be powerfully used. As McAvinney says, while it is possible to specify and manipulate representations of 3D objects with a mouse, decomposing a six degree of freedom task into at least 3 sequential 2D tasks is time-consuming, error-prone and above all *unnatural* [28]. Sculpting or curve specification encounters the same decomposition problem.

A natural way of expressing curves and object edges is to use the fingertip or a pen in 3D space. The 3Draw system, developed at MIT [38] uses a pen with an embedded Polhemus sensor to track the pen's position and orientation in 3D. Another 3SPACE sensor is embedded in a flat palette, representing the plane on which the object rests. The CAD model is thus moved synchronously with the designer's movements. Objects can thus be rotated and translated in order to view them from all sides as they are being created and altered.

Weimer and Gannapathy [43] describe a method of tracing curves in 3D using a VPL DataGlove—shape gesturing as they term it. The fingertip is used to specify the position of a series of control points for a b-spline curve. When a control point is to be selected, isolated speech commands are issued and once recognised, the control point is fixed. Postures could not be used to specify these commands because of the possible ambiguity between commands and curve paths. Also, thumb abduction is used in a clutch and throttle metaphor to transform existing curves.

Telepresence The manual manipulation of robot end-effectors is another physically-based manipulation task that requires concurrent control of multiple degrees of freedom. Conventional robot control devices, levers, dials, joysticks, etc. are inadequate for the simultaneous manipulation of more than three degrees of freedom. In any remote or hostile environment, manually controlled end-effectors may be required to perform complex manipulation. Telepresence is the name given to the area of research that deals with providing the robot‘s teleoperator with as much support as possible for the following problems: visualisation of tasks, the absence of physical feedback and the cognitive problem of mapping hand motion to end-effector motion.

The VIEW system [37] consists of a headmounted stereoscopic display with built-in microphone for speech commands and stereo sound system for audio feedback, a DataGlove to track hand posture for gestured commands, and position and orientation for direct manipulation of objects. Fisher and his colleagues at NASA Ames [11] are developing the VIEW environment to be effective in visualising telepresence applications. Tackling the mapping problem, Pao and Speeter at AT&T [31] have constructed algebraic transformation matrices to map human hand poses to robot hand poses. This transformation matrix is necessary to overcome the kinematic differences between the hand (as represented by a DataGlove) and the Utah/MIT Dextrous Hand.

Interaction & Visualisation 3D visualisation of and interaction with scientific models has been one of the driving problems in this field. Most significant of the scientific visualisation research is project GROPE [4] at University of North Carolina. Since 1967, Brooks and Ooh-Young have been working on a project to develop a tactile and visually interactive display for the 6D force fields of interacting protein molecules. The tactile simulation is explored using a handgrip connected to the Argonne Remote Manipulator which provides force feedback due to the chemical bonding modelled by the computer system. The user, inevitably a chemist, hopes to improve his perception and understanding of the complex force fields caused by molecular bonding. The molecules are manipulated and by minimising the reflected force, the chemist finds a stable bond. Their findings indicate that perception of this cognitively difficult problem does increase and that extremely good bonds can be discovered using the GROPE system.

Medical Studies A glove-like device is a natural tool for a clinical study of the hand's ability to perform physical tasks. In the original VPL DataGlove paper, [45], Zimmerman presents the glove as a hand impairment measuring tool, for patients recovering from strokes, for example. A therapist undertakes the painstaking process using a mechanical goniometer to obtain the range of motion for each individual joint. This process often takes hours to complete. Compare this to the capability of the DataGlove to measure a patient's range of motion for each represented degree of freedom, concurrently, in a fraction of the time and with less qualified supervision necessary. Known tests such as stacking objects or turning over cards and others [17] take a considerable amount of time to perform and equally time-consuming is the analysis of video data captured.

The DataGlove has been used in trials [44] to speed up this process, although given the 5° level of sensor accuracy its usefulness for fine grain measurement is suspect. The Exos DHM, with its sensor accuracy well under 1°, would better facilitate any clinical study than the VPL product. The same process used for rehab studies could easily be applied to degeneracy trials for sufferers of physically debilitating diseases like Parkinson's disease. A quick and often useful metric in these cases is the patient's dexterity in writing their signature. The DataGlove offers another convenient and more powerful metric.

1.3.2 Linguistic Gestures

Sign Language Understanding Understanding a formal signed language, such as the American Sign Language is naturally one of the driving tasks for a project of this kind. The recognition of full, dynamic gestures representing words and concepts as they do in the ASL is undoubtedly the most difficult problem of those mentioned here. There has not, to-date, been any system with these capabilities reported in the literature. There are, however, rumours that companies such as Hitachi and Fujitsu have developed propriortary systems with recognition of several hundred words from a common sign language. Sign language understanding systems have useful syntactic and semantic information available in the context of the sentence being signed. There remains, however, a 60+ degree of freedom[3] problem, incorporating all the problems of gestures performed at different spatial and temporal scales.

The benefits of sign language understanding systems are often debated and not made clear. A functioning system would provide an opportunity for the deaf to communicate with non-signing people without the need for an interpreter—essentially more independence. Although it is argued that a keyboard connected to the speech synthesiser could be used for this purpose, this is not a *natural* interface for the signer and places an intermediary into the dialogue. Another possible problem with a keyboard is that deafness may only be one symptom of physical disability, the user may not have the dexterity to use a keyboard[4].

Another benefit to people who rely solely on sign language would be the ability to converse remotely with others—use a telephone. Videophones may be several years to commercial release, and certainly it will be some time before they are priced within the reach of disability grants. A sign language understanding system could be used to (inexpensively) generate speech or text remotely, or to control a tactile display for the deaf-blind. One inappropriate idea that has been mentioned was to use a SL recognition system to remotely control a multi-jointed artificial arm!

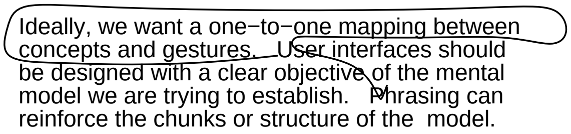
Published research in this area has thus concentrated on finger spelling devices, which are tedious to use for all except proper nouns. Grimes [14] and Kramer [23] developed such systems. Fels [10] went one step further, mapping postures to a vocabulary of root words with the direction and velocity of hand motion providing word modifiers, such as endings and stresses. Tamura

[3] 6 dimensions of wrist pose, 4 joints for each finger and thumb, 2 joints in the wrist, 2 at the elbow and 3 in the shoulder, all doubled for two hand signing.

[4] Randy Pausch's CANDY project [32] aims to develop a speech synthesis controller for Cerebal Palsy sufferers. This is a disease that impairs motor control and although this happens through brain damage, the sufferer is not necessarily mentally disabled, most however, do not have the dexterity to use a keyboard or conventional input device like a joystick. Pausch's system uses the Polhemus 6-dimensional position and orientation tracker on a DataGlove to passively track the movements of some body part which in turn controls an artificial vocal tract. This system should really be mentioned above, since continuous physical movements of the body map onto continuous movements of simulated vocal apparatus.

and Kawasaki [42] report moderate success in recognising a limited high-level gestural language from sequences of video pictures.

Direct Manipulation & Linguistic Interfaces Coleman‘s [8] was the first interface of this type, which allowed a user to edit text interactively using hand-drawn proofreaders symbols to specify commands. Figure 1.1 shows a familiar sight in a proofread text, from Buxton [6]. Giving commands in this way was possible using Coleman's system, returning the word-processor user to a more intuitive and quicker way of editing text. Buxton [7] built a musical score editor



Ideally, we want a one−to−one mapping between concepts and gestures. User interfaces should be designed with a clear objective of the mental model we are trying to establish. Phrasing can reinforce the chunks or structure of the model.

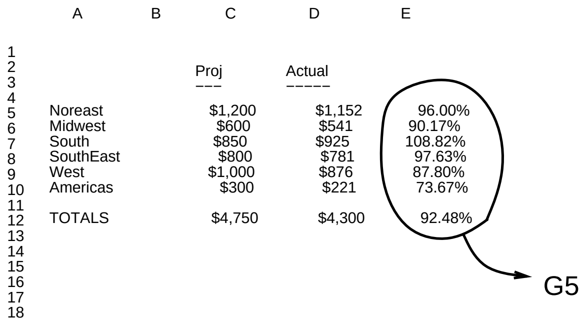
Figure 1.1: Proofreader's Gesture

with gestural input using a mouse. His system used simple gestures to indicate note durations and scoping operations as well as their position on a displayed scale to infer their identity. Rubine [35] developed an application, GSCORE, to do the same thing, while Buxton's technique was incorporated into Notewriter II, a commercial music scoring program.

Minsky [29] developed a visual programming system called Button Box, which uses gestures for object selection, movement and paths of motion. She used a tactile sensing plate mounted on a display as the input device for her system. This plate could sense the position of a finger on the plate and the tactile force. The force sensor was used to distinguish between tapping an object, to activate it, and touching the object, to select it for dragging. Zimmerman [45] describes another ‘visual programming environment', GRASP. Rather than being a *programming environment*, was really the first VPL attempt at building a simulated world populated with objects that could be directly manipulated, by pushing them around, or indirectly manipulated, by pulling postures corresponding to shorthand for pick, move, and so on.

A group at IBM researching human-computer interaction, produced a gesture-based spreadsheet, shown in figure 1.2 from Rhyne and Wolf [34]. The user manipulates cells or groups of cells by gesturing onto a touch tablet, drawing an X to delete, arrows to move and text can be input using hand-writing.

An important development in direct manipulation interfaces was Bolt's "Put-that-There" system published in 1980 [3]. Bolt was the first to combine gesture and speech to tackle the so-called *modality* problem. This refers to the difficulty of specifying several simultaneous parameters unambiguously. The concept of a mode in an application is common, MacDraw, for example, may be in one of a



	A	B	C	D	E
1					
2			Proj	Actual	
3			---	-----	
4					
5	Noreast		$1,200	$1,152	96.00%
6	Midwest		$600	$541	90.17%
7	South		$850	$925	108.82%
8	SouthEast		$800	$781	97.63%
9	West		$1,000	$876	87.80%
10	Americas		$300	$221	73.67%
11					
12	TOTALS		$4,750	$4,300	92.48%
13					
14					
15					
16					
17					
18					

Figure 1.2: A Gesture-based spreadsheet

number of modes, including line, circle and text. The next operation performed produces the corresponding object. Mistakes occur in this sort of application when the user believes he is in one mode but is actually in another. Allowing the specification of several parameters simultaneously eliminates the need for modes. This would be the case if, in MacDraw, the user could specify simultaneously that a line should be drawn and the start and end coordinates. Rubine‘s [35] GDP (gesture drawing program) is similar to MacDraw but instead of a palette of tools, and different modes, the shape and position of a gesture performed using a mouse, determines that of the object.

In “Put-That-There”, [3], Bolt ordered data around a large projected graphical display in several ways. Pointing, the position and direction of which is tracked by a Polhemus 3SPACE, was used to disambiguate objects chosen with spoken pronouns. “Put that ...”, the object would dissolve to let the user know it had acknowledged the reference, “...there”, pointing at another position on the screen. Pure speech commands were also possible, “Put the green triangle to the left of the yellow square”.

Many subsequent systems [37, 43, 1] have used speech to validate objects chosen by direct manipulation, or to eliminate the ambiguity of gesturing in both ‘modes’ that were isolated in the opening section, 1.1, *physical* and *linguistic*.

1.4 Gesture Recognition at TCD.

The Computer Vision Group of Trinity College is developing a gesture recognition system as part of its work in the ESPRIT II Research Project (GLAD-IN-ART).

After a thorough survey of existing gesture recognition systems, it was decided that if the system developed here was to meet its goals, a novel approach was necessary. No existing system recognises *full, dynamic gestures.* The system developed here promises, by its method of representation to recognise dynamic gestures by modelling the change of degrees of freedom over time.

The currently used methods in gesture-based systems are neural network matching, matching templates to input data and statistical classification. The first two methods, at least in the way they are conventionally presented, do not handle time-variance well. They are more suited to static patterns, postures, for example. The features extracted from the input to be used by statistical classifiers may include time, but as section 2.4 will show, picking the relevant features to extract is often dependent on the actual gesture pattern. The aim of this project was to remove this specifity and develop a method that will work for simple or complex gestures. Other methods which are used for matching time-space curves, spline models, for instance, are too expensive computationally. The recognition of gestures must be performed in *real time* and a standard spline approach would not be possible on the available hardware.

Practically, the main motivating factor for this system was its involvement in the GLAD-IN-ART[5] consortium. This is an ESPRIT II funded project investigating the problem of implementing the direct manipulation of simulated or virtual entities. Central to this project are the development of new posture and pose tracking systems, with which the interaction between the user's hand and graphical objects can be accurately modelled. Although this direct interaction is the main focus of the project, it is acknowledged that indirect manipulation of objects and the environment is a powerful method on human-machine interaction. Thus the gesture recognition system described in this report is an important part of the overall system. Figure 1.3 shows the main components of the GLAD-IN-ART architecture.

The recognition system, however, is not dedicated to the GLAD-IN-ART project alone. It provides a clean interface to applications which may wish to use gestural input. The system receives a list of gesture identifiers and their corresponding templates on initialisation and places no semantic meaning on the recognised gestures. This allows applications to use gestures where appropriate. This emphasis on a clean interface to a client program is precisely what is needed for X Windows applications. Thus this system is ideal for integrating into X-based applications.

Another motivation for this project was as an investigation into Human-Computer Interaction, Pattern Recognition and their intersection. This chapter has provided an overview of the history and state-of-the-art of gesture-based HCI. The next chapter looks at alternative techniques for recognising patterns, particularly how researchers have recognised gestures as patterns.

[5] Glove-like Advanced Interface for the Control of Manipulatory and Exploratory Procedures in Artificial Realities

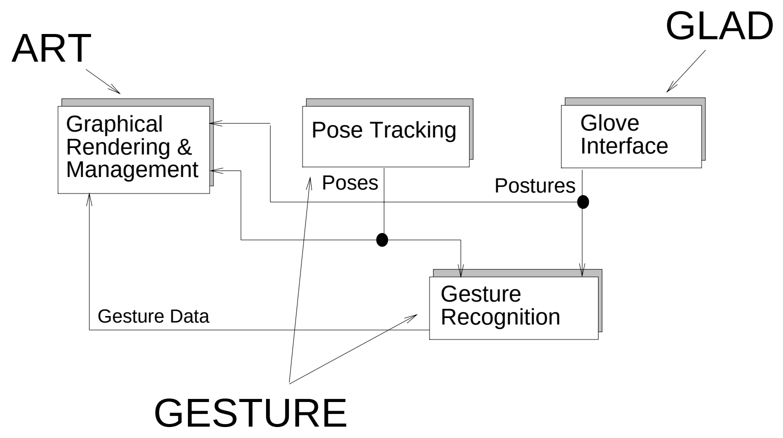


Figure 1.3: The main modules of the GLAD-IN-ART consortium. Trinity College is responsible for the Pose tracking and Gesture Recognition modules.

Chapter 2

Approaches to the Problem

Fu [13] states that

> The problem of pattern recognition usually denotes a discrimination or classification of events.

Clearly, gesture recognition may be viewed as a problem of pattern recognition, in which the events to be classified are instances of input from posture and pose sensors.

In general, pattern recognition (PR) consists of two subproblems: pattern representation and decision making. Thus, formally, the architecture of any approach to PR consists of two subsystems. The **representer** takes the raw pattern, which might be an image, a string, or in this case a timestamped stream of sensor data, and outputs its internal representation of the pattern. This internal representation, often a set of features extracted from the data, is in the most convenient form for the **decider**, to take as input and hence output a classification for the pattern, (if one exists).

This chapter concentrates on possible PR techniques that could be used to solve the problem of gesture recognition. It focusses particularly on techniques used by researchers in the past, their relative successes and scope for flexibly recognising full gestures. Gesture recognition is distinct from posture recognition in that it requires the recognition of hand shape in time and space domains, rather than only the space domain. The techniques will be examined for their ability to deal with spatial and temporal scaling and for their training procedures.

2.1 First Steps

Probably the earliest posture recognition was by Grimes [14] whose Digital Data Entry Glove was designed specifically for recognising the signed alphabet.

Carefully placed sensors registered fingertip contact, flexion of specific joints, and hand attitude. Posture recognition was hard-coded into the electronics of the glove; a particular combination of sensor readings produced an 'A', another a 'B' and so on. The advantage of this technique is rapid and robust posture recognition. The disadvantage is inflexibility, in that only those postures it was designed for can be recognised. Neither Grimes nor Sturman [41] mention whether calibration was used to adjust for each user before postures were recognised.

In 1980, Bolt used a prototype Polhemus 6D pose tracking system for sensing the direction of a point in his 'Put-that-there' system [3], described in section 1.3.2. The sensor was attached to the user's wrist and pose information enabled the system to (at least) estimate where he/she was pointing at.

2.2 Template Matching

Template matching is probably the simplest method of recognising gestures to be employed. Essentially there is no representation stage, the 'raw' sensor data is used as input to the decider which is based on a function which measures the similarity between templates of values and the input. The input is classified as being a member of the same class as the template to which it is most similar, or nearest, looking at the classification stage as a map from the input to joint space in which the templates occupy points or subspaces. In general, there is a similarity threshold, below which the input is rejected as belonging to none of the possible classes (or too far from the nearest template in joint space). In fact the similarity function is most often the Euclidean distance between the set of observed sensor values and a template.

2.2.1 VPL's Posture Recognition

VPL's ground breaking work in 1987 [45] used a method for recognising postures which followed very closely to that outlined above. For each posture each sensor had a range of values that were valid. At each sample time[1], the sensor readings were compared with the values of the posture templates. The absolute value of the difference for each sensor is summed for each template. The gesture with the minimum sum, below a global threshold, was the one chosen.

VPL's software also provided hysteresis values for each sensor value to widen the range of the match once a posture has been recognised, helping the user to hold a posture after recognition. [45] also describes a simple calibration technique to allow the ranges to be altered to suit different users of the DataGlove, since the sensors would be in slightly different positions for each users hand.

The advantages of the VPL approach is that it is simple enough to generate and recognise postures efficiently, the representation of postures is easily under-

[1] Approximately $\frac{1}{60}$ second for the VPL DataGlove

stood and modifiable by the user using a *gesture editor* provided. However, in practice, the range of each template entry must be quite wide, estimated at up to 30% of the total flexion range of each sensor. This is due partly to inaccuracies in sensing and partly to misperformed postures. Hence with more than about ten to fifteen postures, the templates begin to overlap in joint space.

Lipscomb [26] also used a template matching based method for recognising used for recognising stroke, i.e. 2D, gestures. This was a variant of the usual technique where multiple templates were maintained for each gesture, corresponding to increasingly coarse resolutions of sensor values[2]. The templates were examined first at the lowest resolution and only if successful at this stage, would the template proceed to matching at a higher resolution.

2.2.2 Extending Posture templates to Gestures

To approach gesture recognition using a template matching scheme, gestures would have to be recognised as sequences of postures. In this technique, trajectories of the degrees of freedom are not modelled, hence spatial scaling would be impossible, and temporal scaling although possible, would be somewhat inflexible.

2.2.3 Syntactic Classification

In an approach similar to that taken by Fu [12] for several applications, Jones and Doyle [19] suggest an alternative to sequentially calculating the similarity measures for each template, given a set of observed values.

The method is based on constructing and traversing a *feature tree*, like the one partially shown in figure 2.1. Key features are represented by sub-branches, and as the observed values are examined, a tree traversal takes place where, for example, if the user's index finger is extended the traversal first takes the left sub-branch. All of the gestures with this feature would be represented in this sub-branch. The features are not mutually exclusive, so that, the same gesture could be represented in more than one place in the tree. The organisation of the tree, that is the features encoded as the lexemes, and the order of traversal would present a problem, as would addition or removal of gestures, since the tree would have to be regenerated for efficient traversal.

In the last paragraph, the features of a posture were represented as lexemes grouped to make a word (or posture), an extension of this method could be used to recognise sequences of postures as sentences. Figure 2.2 shows an example of this extension.

Linguistic techniques, apart from those used by Fu, are relatively common, particularly so in visual pattern recognition. Two examples are described in Rubine [35]:

[2] Used in a similar manner to the way in which a Laplacian of Gaussian filter is used in Image processing.

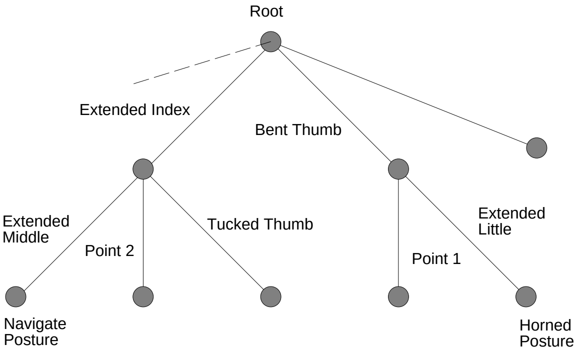


Figure 2.1: A partial tree of postural features

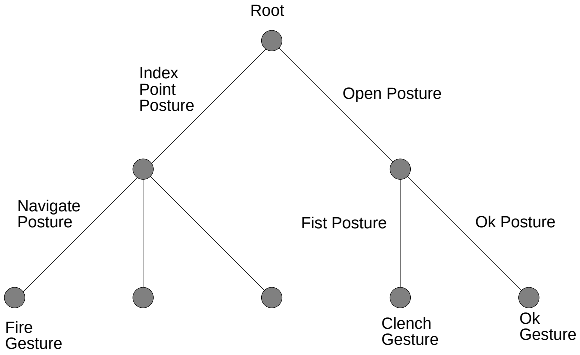


Figure 2.2: A partial tree of posture sequences leading to gestures

- Shaw's picture description language (PDL) has been successfully used to describe and classify line drawings, [39].

- Stallings uses the composition of operators *left-of*, *above* and *surrounds* to describe the relationships between strokes of Chinese characters, [40].

Critique Finally, a critique of the template matching technique is given.

These techniques are easy to develop and computationally non-complex. There are theoretical problems with the use of templates, however. How, for example, should the templates be constructed, given examples of gestures or postures, and how should they be adaptive: altered to suit the needs of individual users, independent of sensor calibration schemes. Template matching, then, does not have the formal approach to training that the two following methods, neural networks and statistical classifiers have.

2.3 Neural Networks

Neural networks have received much attention for their successes in pattern recognition. Gesture recognition is no exception to this and several systems have been reported in the literature.

Informally, the reason for their popularity is that once the network has been configured, it forms appropriate internal representer and decider systems based on training examples. Also, because the representation is distributed across the network as a series of interdependent weights instead of a conventional *local* data structure, the decider has certain advantageous properties: recognition in the presence of noise or incompleteness and pattern generalisation. Generalisation plays a crucial role in the system's performance, since most gestures will not be reproduced even by the same user with perfect accuracy, and when a range of users are allowed to use the system, the variation becomes even greater. Other useful properties of this approach include performing calibration automatically and the ability to classify 'raw' sensor data, as with template matching.

However, neural networks have very serious drawbacks:

- Often thousands of labelled examples are needed to train the network for accurate recognition.

- This training phase must be repeated from the start if a new gesture is added or one is removed.

- Neural networks can *over learn*, if given too many examples, and discard originally learnt patterns. It may also happen that one bad example may send the patterns learnt in the wrong direction, and this, or other factors, such as orthogonality of the training vectors, may prevent the network from converging at all.

- They tend to consume large amounts of processing power, especially in the training phase.
- The biggest problem is to understand what the network had actually learnt, given the ad-hoc manner in which it would have been configured. There is no formal basis for constructing neural networks and the topology, unit activation functions, learning strategy and learning rate all must be determined by trial and error.

Despite these drawbacks, somewhat successful gesture recognition has been performed with neural networks, notably by Fels [10].

2.3.1 Fels' GloveTalk

His work concentrated on building a gesture-to-speech interface, using a VPL DataGlove connected to a DECtalk speech synthesiser via a series of neural networks. The neural networks that adapt the mappings for a user from training examples of gestures must be small to render the system computationally feasible. By dividing the task functionally into 5 independent networks this has been achieved:

1. hand-trajectory—strobe time network, determines when a gesture has been made
2. hand-shape—root word network, determines which root word has been made from posture
3. hand-direction—word ending network, determines word ending from direction of hand motion
4. hand-displacement—word stress network, determines word stress from displacement from initial position
5. hand-speed—word rate, calculates rate of required speech from the speed of hand motion

As can be seen from above, posture and pose are considered separately, and posture provides the 'stem' from the vocabulary and position, provides the parameters, word ending, stress and speech rate. Thus, the system is essentially a posture recognition system, since finger motion is not modelled, and hand motion consists only of variable speeds back and forth.

The neural networks were implemented using the Xerion Simulator developed by Hinton at the University of Toronto. The back-propagation of errors method [36] of supervised learning was used, and typically for this type of learning the training data required was input vectors paired with target output vectors. Fels admits that 30356, *hand-labelled*, input/output pairs were needed to train the network adequately. Once trained, the network recognised robustly, Fels' reported a 92% success rate on the recognition of 203 signs based on 66 hand-shapes combined with 6 movement stages.

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

2.3.2 Brooks' Kohohen Net approach

Brooks [5] also reports use of a neural net to control a mobile robot by interpreting DataGlove motion. In ways, this surpasses Fels' work, particularly by incorporating dynamic gestures into the system's vocabulary. The Kohonen net model [21] is employed to recognise paths traced by degrees of freedom in n-dimensional space. Each Kohonen net, typically as small as 20 units, was trained to recognise a single gesture. Concurrent operation of the networks could classify the input paths into known classes with moderate success.

The system reported on was certainly an early prototype since very simple trials had been conducted such as closing all fingers, leading with index; opening the thumb and first two fingers simultaneously; and moving from a neutral posture to a 'pen' grasp posture. Brooks concluded that he has yet to show that his methods are sufficient for practical dynamic gesture recognition [3].

2.3.3 Beale and Edwards' Postures

In [2], Beale and Edwards use a multilayer perceptron model [36] to classify input into one of five postures, taken from the American Sign Language. The network had ten input units, one for each angular degree of freedom sensed in a VPL DataGlove, and five output units, one for each of, *a, e, i, o* and *u*. The number of units in a single hidden layer, originally tested at eight was empirically pared down to three.

The required output for positive recognition was strong output from one of the units, and practically zero response from the others. Interestingly, the learning rate, η, and the network momentum term, α, had a negligible effect on the final effectiveness of the network. This would seem to indicate that the *energy landscape*[4] for the data set is relatively simple and unconvoluted, which in turns would suggest that the learning task is straightforward. The

[3] Kohonen nets are more formally based on linear algebra than, for example, the PDP models, and this implies strict algebraic relationships between the patterns being learned. Much analysis would have to be carried out to ensure the gestural patterns were algebraically suitable for training.

[4] Stated simply, this is the pattern of network weights, pictured as a 3D terrain map.

researchers statistics show that learning time, (for just five postures), is of the order of minutes.

As with the present project, a glove-like device was not available so DataGlove data was simulated. The simulation was performed not using a package, but simply by preparing the input vectors numerically. Beale and Edwards report successful matching up to noise levels of 20%, and quote Quam et. al. [33], in which they report a 10% error rate covers most users of sign language, to show that the recognition is sufficiently robust.

A serious criticism of this work must be the number of postures covered. With five, unsimilar postures, problems of overlapping classes will not occur, an issue that needs to be addressed.

A method for using neural networks to recognise gestures is also mentioned. The approach uses recurrent networks, by Jordan [20], whose architectures encode the temporal information about prior network states.

2.4 Statistical Classification

In statistical matching, the statistics of example feature vectors are used to derive deciders, usually called classifiers. Functionally, statistical classifiers operate in the same way as either of the previous methods—mapping an n-feature vector to a point in n-space, where it belongs to one of a number of classes. The mapping function, however, uses statistical decision theory, Bayesian maximum likelihood theory, for example, to decide on the class membership given a point in feature space. The features extracted in the representation phase, are most often hand-picked for their particular relevance to the application.

2.4.1 Rubine's stroke recognition

The most important work on gesture recognition using statistical classification is that by Rubine [35]. For his PhD thesis, Rubine created not only a gesture recognition system, but GRANDMA[5], an object-oriented toolkit for building gesture-based applications. The gestures considered in his work consist of the two dimensional path of a single point over time. These are gestures that may be input with a single pointer, such as a mouse, stylus or touch pad. The term *gesture* is used in the following in Rubine's sense. Certain limiting constraints are assumed, notably that the start and end of a gesture are clearly delineated. The start of a gesture might be, for example, indicated by the depression of a mouse button and the end by its release. Each gesture is an array g of P timestamped sample points:

$$g_p = (x_p, y_p, t_p) \quad 0 \leq p < P$$

[5] Gesture Recognisers Automated in a Novel Direct Manipulation Architecture

Let C be the number of gesture classes, each specified by example gestures. Given an input gesture, g, the recognition problem is stated as follows: to determine the class, c, to which g belongs.

2.4.2 The features

Rubine labels his vector of features, $\mathbf{f} = [f_1, \ldots, f_F]$, geometrically based on the path of the gesture, and carefully computed incrementally. With reference to figure 2.3, the following are examples of some of the 13 features chosen by Rubine : f_1 and f_2, the cosine and sine of the initial angle of the gesture, the length, f_3, and the angle, f_4, of the bounding box diagonal and f_5, the distance between the first and last points.

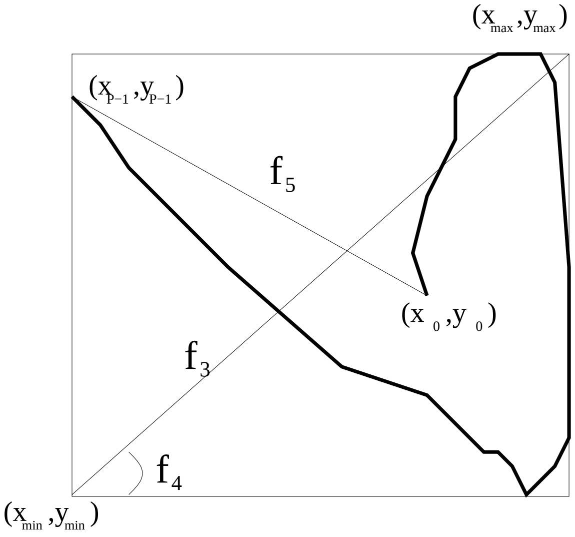


Figure 2.3: A gesture and example feature points extracted by Rubine's classifier

$$f_1 = cos\alpha = \frac{(x_2 - x_0)}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}}$$

$$f_2 = sin\alpha = \frac{(y_2 - y_0)}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}}$$

$$f_3 = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2}$$

$$f_4 = arctan\frac{y_{max} - y_{min}}{x_{max} - x_{min}}$$

$$f_5 = \sqrt{(x_{P-1} - x_0)^2 + (y_{P-1} - y_0)^2}$$

Rubine himself says

> The aforementioned feature set was empirically determined by the author to work well on a number of different gesture sets.

2.4.3 Classification

Once the feature vector, f is computed for an input gesture, g, a linear evaluation function, one for each of the classes, is computed over the features, f. Gesture class c has weights $w_{\hat{c}i}$ for $0 \leq i < F$, where F is the number of features. This set of weights is derived by a training a linear classifier with the example gestures. Rubine reports that typically 15 examples are sufficient to train the classifier. The evaluations, somewhat equivalent to computing Euclidean distance in template matching, or measuring output unit response in a neural network, are made:

$$v_{\hat{c}} = w_{\hat{c}i} + \sum_{i=1}^{F} w_{\hat{c}i} f_i \quad 0 \leq c < C$$

The classification of g is simply the c which maximises $v_{\hat{c}}$. Rubine reports the classification rates for different types of gestures as: Simple Shapes, 100%; Digits, 98.5% and Letters, 97.1%.

2.4.4 Extending the System

Rubine also explains how his system would cope with multi-path data, such as that produced by a glove-like device. By treating the multiple degree of freedom input as paths, e.g. the paths of the fingertips, the single stroke recognition algorithm may be applied to each of the paths individually and the results combined to classify the gesture. This combination of results is implemented using a syntactic technique based on a tree of possible classifications from the individual paths. However, the start and end of the paths still have to be made explicit—a somewhat artificial constraint.

Sturman [41] extended Rubine's systems to deal with multi-path gestures using a VPL DataGlove. Significantly, the feature analysis was extended to three-dimensions and modified to permit continual analysis and recognition without explicit start and end points. As in Rubine's work, the feature set was chosen empirically to be useful for recognition of specific gestures.

However, the interpretation of features was performed using an explicit formulation for each gesture, rather than by means of a generic pattern recognition algorithm as used by Rubine. Sturman says

> Advantages of explicit formulation are that gesture recognition routines need no training or samples from individual users. If properly formulated, the routines will work for all users and efficiently

use only relevant features. Disadvantages include that the user may need minor training to produce some of the gestures, and new gestures require new formulations.

Apart from the slight contradiction concerning the need for training, Sturman's approach seems most inflexible: he chooses the features to analyse in each gesture and how to analyse them, individually.

For example, his formulation of a 'waving' gesture depends on the following conditions:

1. confirm that the hand is not closed by using the simple posture recognition of making sure the MCP[6] joints of the four fingers are less than 0.8 flexed.

2. look at a feature that counts the number of direction switches in a single valued variable over the last N updates. If this value is less than 5 for any of the MCP joints over the last N updates, then waving is not detected.

3. check that the waving motion is large enough to avoid catching small random motions of the hand. This is accomplished by accumulating the path segment lengths through a linear filter:

$$y_{i+1} = kx_i + (k-1)y_i \quad k < 0.5$$

   This removes input samples more than N frames old from the accumulated value. If the value is not within a chosen range for any one of the four MCP joints, waving is not detected.

The conditions for Sturman's other gestures are similarly specific.

This approach means that new gestures could not be added by users as they would require new formulations. However, this is not necessarily a negative criticism, since Sturman's system was more a feasibility study of the area of whole-hand input, than a 'product'.

2.5 Discontinuity Matching

The approach taken in the GLAD-IN-ART project is a novel one. The **classifier** is a classic template matcher, thus the novelty occurs in the feature extraction. Consider the movement of a finger joint over time; figure 2.4 shows how, for example, an MCP joint changes as the posture changes from an open hand to a clenched hand. The features that are represented are the critical points of the time-space pattern. This first derivative representation is reminiscent of a spline approximation to a curve, but without the intermediate knots, (control points).

[6] Metacarpophlangeal Joints, the ones at the knuckles.

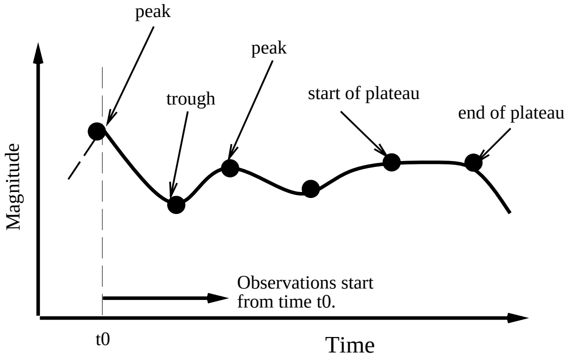


Figure 2.4: Time-space pattern of an MCP joint in performing a clench gesture

The advantage this approach has over the classic template matching approach is that the relevant features are motion oriented, this is necessary for a system that performs gesture recognition. Also, since the discontinuities need not be matched at precisely the same points in time or space, the system is robust to scaling. The other advantage of representing merely critical points is that the system is less susceptible to input noise. Using a simple filter, jitter can be subtracted from the sensor input. What remains are gross discontinuities, which are unlikely to be mis-classified by the representer.

2.6 The related problems of handwriting and speech recognition

The problems of handwriting and speech recognition are closely related to that of recognising gestures, all three may be viewed from a signal processing point of view as analysis of a time-value curve. Gesture recognition, however, must be considered the more difficult problem since many more degrees of freedom are involved.

In terms of techniques used, there is about the same mix of statistical, connectionist and template based classifiers as described above, but the where the methods are differentiated is in their feature extraction. All of the methods for solving these problems use special purpose feature extraction modules based on the type of filters and preprocessors used on the raw data. These preprocessors extract features relevant to the task, features which have been researched for many years. This is the real difference between Speech and Handwriting on one side and gesture recognition on the other: there does not exist, for gestures, a significant body of research that points the feature extraction techniques towards relevant features. As described above, the features used are chosen empirically for their usefulness.

> The most important problem in handwriten character recognition theory is that of extraction of features which are hardly affected by handwriting distortions.
>
> Shozo Kondo[7] [22].

In the recognition of speech, the same is true: as Lea reports in [25], that many of the features of the speech waveform that are worth considering are at higher levels where they are less suceptable to noise. The technique of Dynamic Programming is used liberally in speech recognition to warp the time scale of observed data to the templates of stored sounds. These stored sounds are most commonly based at sub-syllable or phoneme level.

Another very serious issue that an examination of the techniques of speech and script recognition uncovers, is that of context. The use of contextual information in these two problems greatly simplifies the task of continuous recognition.

To illustrate this point, Lea presents four possible viewpoints to a speech recogniser:

1. Signal Processing—approach the speech signal without any context.

2. Speech Production—understand the 'source' of the signal, model essential aspects of the way in which the human vocal system produces speech: vocal chord resonances, for instance.

3. Sensory Perception—duplicate the human auditory response process.

4. Speech Perception—make categorical distinctions that are experimentally important, high level feature extraction.

This list of approaches ranges from the 'ignorance model' to the 'knowledge-source-driven model'.[8]

It seems that gesture recognition is being approached at the 'ignorance' level. The reason for this is that in relation to the other two problems, gesture recognition is a very young problem—the knowledge has not been accumulated. Zimmerman in his seminal paper [45] acknowledges

> As in speech recognition, dynamic gesture recognition is able to take advantage of context in order to limit the number of gestures to be distinguished at a given time.

[7] It is no coincidence that the vast majority of work in the area of recognition of handwriting has been conducted by oriental researchers. While arabic scripts are more efficiently typed than written, the Kanji scripts used in Chinese and Japanese langauges are very cumbersome for keyboards.

[8] Newell's terms.

It remains to be discovered what exactly is the *context* of gestures, undoubtedly this will depend on the application. The context of sign langauge, for example, would be syntactic and semantic information in the signed sentence, along with facial expression and body movement.

Bibliography

[1] Enrico Balaguer and Jean-Francis Gobbetti. Virtuality builder ii: On the topic of 3d interaction. Technical report, EFPL, Swiss Federal Institute of Technology, Lausanne, 1993.

[2] R Beale and A Edwards. Recognising postures and gestures using neural networks. In R. Beale and Finlay J., editors, *Neural Networks and Pattern Recognition in Human Computer Interaction*. E. Horwood, 1992.

[3] Richard A. Bolt. "put-that-there": Voice and gesture at the graphics interface. *Computer Graphics*, 14, No. 3:262–270, July 1980.

[4] Frederick P. Brooks et al. Project grope—haptic displays for scientific visualisation. *ACM Computer Graphics*, pages 177–185, 1990.

[5] Martin Brooks. The dataglove as a man-machine interface for robotics. In *The Second IARP Workshop on Medical and Healthcare Robotics*, Newcastle upon Tyne, UK, September 5-7 1989.

[6] W.A.S. Buxton. There's more to interaction than meets the eye: Some issues in manual input. In D.A. Norman and S.W. Draper, editors, *User Centred Systems Design: New perspectives on HCI*, pages 319–337. Lawrence Erlbuam Associates, 1986.

[7] W.A.S. Buxton et al. The evolution of the sssp score-editing tools. In Curtis Roads and John Strawn, editors, *Foundations of Computer Music*, chapter 22, pages 433–466. The MIT Press, 1985.

[8] Michael Coleman. Text editing on a graphic display device using hand-drawn proofreader's symbols. In *Pertinent Concepts in Computer Graphics, Proc of the Second University of Illinois Conf on Computer Graphics*, pages 283–290, 1969.

[9] Howard Eglowstein. Reach out and touch your data. *Byte*, pages 283–290, July 1990. Review of three commercial hand tracking devices.

[10] S. Sidney Fels and Geoffrey E. Hinton. Building adaptive interfaces with neural networks: The glove-talk pilot study. In *Human-Computer Interaction—INTERACT 9́0*, pages 683–688. IFIP, Elsevier Science Publishers B.V. (North-Holland), 1990.

[11] Scott S. Fisher. Telepresence master glove controller for dexterous robotic end-effectors. In *SPIE Vol. 726, Intelligent Robots and Computer Vision: Fifth in a series*, 1986.

[12] K.S. Fu. *Syntactic Pattern Recognition.* Prentice-Hall, Inc., 1981.

[13] K.S. Fu and T.S. Yu. *Statistical Pattern Classification using Contextual Information.* Recognition and Image Processing Series. Research Studies Press, 1980.

[14] Gary J. Grimes. Digital data entry glove interface device. Technical Report US Patent 4,414,537, Bell Telephone Laboratories, November 1983.

[15] Rheingold Howard. *Virtual Reality.* Secker & Warburg, 1991. Exploring the brave new technologies of artificial experience and interactive worlds from cyberspace to teledildonics.

[16] Edwin L. Hutchins et al. Direct manipulation interfaces. In D.A. Norman and S.W. Draper, editors, *User centred system design*, pages 87–124. Lawrence Erlbaum Associates, Inc., 1986.

[17] R.H. et al. Jebsen. *Archives of Physical Medicine and Rehabilitation*, 50:311–316, 1969.

[18] Peggy J. Jennings and Howard Poizner. Computergraphic modelling and analysis ii three-dimensional reconstruction and interactive analysis. *Journal of Neuroscience Methods*, 24:45–55, 1988.

[19] Doyle R Jones M and P O'Neill. The gesture interface module. GLAD-IN-ART deliverable 3.3.2., Trinity College, Dublin, January 1993.

[20] M Jordan. Serial order: A parallel distributed processing approach. Technical Report 8604, Institute for Cognitive Science, University of California, San Diego, 1986.

[21] Teuvo Kohonen. *Self-organisation and associative memory.* Springer-Verlag, 1984.

[22] Shozo Kondo. Automatic recognition of handwriting. In C. Suen et al., editors, *Computer Reocgnition and Human Production of Handwriting.* World Scientific, 1989.

[23] J. Kramer and L. Leifer. The talking glove: An expressive and receptive 'verbal' communication aid for the deaf, deaf-blind and nonvocal. In *Proc. Ann Conf Computer Tech/Spec Ed/Rehab*, pages 335–340, Conference held at California State University, Northridge, CA, 1987.

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

[25] Wayne Lea. Components of a speech recognizer: Past and present. In W.A. Lea, editor, *Trends in Speech Recognition*, Signal Processing Series. Prentice-Hall, Inc., 1980.

[26] J.S. Lipscomb. A trainable gesture recogniser. *Pattern Recognition*, 1991.

[27] Beth A. Marcus and Philip J. Churchill. Sensing human hand motions for controlling dexterous robots. In *The second Annual Space Operations Automation and Robotics Workshop, held at Wright State University*, July 1988. Sponsored by NASA and the USAF.

[28] Paul McAvinney. Telltale gestures. *Byte*, pages 237–240, July 1990.

[29] Margaret Minski. Manipulating simulated objects with real-world gestures using a force and position sensitive screen. *Computer Graphics*, 18(3):195–203, July 1983.

[30] Paul O'Neill. Hand pose from iconic markings. GLAD-IN-ART deliverable 3.1.3., Trinity College, Dublin, 1993.

[31] Lucy Pao and Thomas H. Speeter. Transformation of human hand positions for robotic hand control. In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, pages 1758–1763, 1989.

[32] Randy Pausch and Ron Williams. Giving candy to children: User tailored gesture input driving an articulator based speech synthesiser. *Communications of the ACM*, 35(5):59–66, May 1992.

[33] D. Quam et al. An experimental determination of human hand accuracy with a dataglove. In *Proc. Human Factors Society 33rd Annual Meeting*, volume 1, pages 315–319, 1989.

[34] James Rhyne and Catherine Wolf. Gestural interfaces for information processing applications. Technical Report RC12179, IBM T.J. Watson Research Centre, September 1986.

[35] Dean Rubine. *The Automatic Recognition of Gestures*. PhD thesis, Carnegie Mellon University, December 1991.

[36] D.E. Rumelhart, J.L. McClelland, et al. Parallel distributed processing: Explorations in the microstructure of cognition. volume 1: Foundations., 1986.

[37] Fisher Scott S. et al. Virtual interface environment workstations. In *Proceedings of the Human Factors Society — 32nd Annual Meeting*, 1988.

[38] Emmanuel Sachs. Coming soon to a cad lab near you. *Byte*, pages 238–239, July 1990.

[39] A.C. Shaw. Parsing of graph-representable pictures. *JACM*, 17, 1970.

[40] W.W. Stallings. Recognition of printed chinese characters by automatic pattern analysis. *Computer Graphics and Image Processing*, 1, 1972.

[41] David J. Sturman. *Whole Hand Input*. PhD thesis, Massechusetts Institute of Technology, 1992.

[42] Shinichi Tamura and Shingo Kawasaki. Recognition of sign language motion images. *Pattern Recogntion*, 21(4):343–353, 1988.

[43] David Weimer and S.K. Ganapathy. Interaction techniques using hand tracking and speech recognition. In Meera Blattner and Roger Dannenberg, editors, *Multimedia Interface Design*, Frontier Series, pages 109–126. Addison-Wesley Publishing Company, 1987.

[44] Sam et al. Wise. Evaluation of a fibre optic glove for semi-automated goniometric measurements. *Journal of Rehabilitation Research and Development*, 27(4):411–424, 1990.

[45] Thomas G. Zimmerman and Jaron Lanier. A hand gesture interface device. *ACM SIGCHI/GI*, pages 189–192, 1987.