Vision Based Dynamic Hand Gesture Recognition for Human Computer Interaction

A
Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
Doctor of Philosophy

By
Manas Kamal Bhuyan

Lakshminath Bezbaroa Central Library
IIT Guwahati

Department of Electronics and Communication Engineering
Indian Institute of Technology Guwahati
Guwahati-781039, India.
September, 2006.

THESIS
Lakshminath Bezbaroa Central Library
Indian Institute of Technology Guwahati
ACC. No. TH. 1870
Date..... 28/3/19.....

621.3
3110/N
P06

TH-1870_02610203

I would like to dedicate this thesis to
my wife Mizomi,
daughter Jia,
and my parents.

TH-1870_02610203

Acknowledgements

I would like to take this opportunity to express my appreciation to all the people who have guided, supported, and encouraged me in carrying out this thesis work. I would like to thank my supervisors Dr. Debashis Ghosh and Prof. Prabin Kumar Bora for their guidance and constant support during this research. I feel it is a great privilege to express my deepest and most sincere gratitude to them for their suggestions, constant encouragement and their valuable comments on my work. I also thank my Doctoral Committee members, Prof. A. Mahanta, Dr. J.S. Sahambi and Dr. S.V. Rao, for their invaluable comments from time to time that have enriched this thesis work. The cooperation I received from other faculty members of the Department is gratefully acknowledged.

I thank my lab colleagues, other research scholars and all staffs of the Department for their help and support in carrying out my experiments and other thesis related works. I specifically thank Mrinal and Vinod, my closest partner, who have been particularly cooperative and supportive throughout this course of work.

Of course, I am grateful to my wife Mizomi and my daughter Jia for their patience and *love*. Without their inspiration and moral support, this work would never have come into existence (literally). Finally, there are others who have influenced this research indirectly, but fundamentally, through their influence on my life. They are my parents, in-laws, sister and brother, whose love, patience, and encouragement made this research possible. Last but not least, the encouragement and motivation that was given to me to carry out my research work by Late Joytimoy (Lathu) is also remembered. THANKS!!!

IIT Guwahati, India.
Sept 14, 2006

Manas Kamal Bhuyan

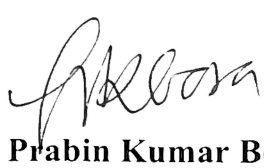
TH-1870_02610203

Certificate

This is to certify that the thesis entitled "**Vision Based Dynamic Hand Gesture Recognition for Human Computer Interaction**", submitted by Manas Kamal Bhuyan, a research student in the *Department of Electronics and Communication Engineering , Indian Institute of Technology, Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under our supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the Institute and in our opinion it has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.



Dr. Debashis Ghosh.



Prof. Prabin Kumar Bora.

Department of Electronics and Communication Engineering,
Indian Institute of Technology Guwahati,
Guwahati-781039,
India.

TH-1870_02610203

Abstract

With the increased interest in human-computer interaction (HCI), there has been rapid growth of research related to gesture recognition in recent years. Hand gesture recognition from visual images forms an important part of this research. This thesis reports on our research on recognition of dynamic hand gestures having different spatio-temporal characteristics. We develop methods to recognize dynamic hand gestures with (1) local hand motion only where only the fingers and the palm move without any movement of the whole hand, (2) global motion only where the hand as a whole moves in space to make different gestures, and (3) both local and global motions where the fingers and palm create different hand poses as the arm traverses along a trajectory in space.

We use the concept of object-based video abstraction for segmenting the frames into video object planes (VOPs) with each VOP corresponding to one semantically meaningful hand pose or position in space. A gesture is represented as a sequence of key video object planes (key VOPs) that correspond to significantly different VOPs in the video sequence. A local-motion gesture is modelled as a sequence of states, each state corresponding to the hand shape in each of the key VOPs. The state along with the temporal information in terms of the number of VOPs between two consecutive key VOPs forms a finite state machine (FSM) for the gesture. Recognition of this type of gesture is performed on the basis of FSM matching. For gestures with global motion only, gesture trajectories obtained from the key VOPs are used for recognition via trajectory matching under time normalized condition. To accomplish trajectory guided recognition, we propose to use some basic static and dynamic trajectory features. In case of gestures having both local and global motions, recognition is accomplished by combining trajectory matching with the FSM-based hand pose matching. For recognition of a continuous stream of gestures, a gesture is discriminated from the movement epenthesis. A movement epenthesis phase due to local hand motion is modelled using the FSM-based representation and that due global hand motion is represented by fuzzy modelling of hand motion parameters. Experimental results show that the proposed gesture recognition methods perform well in recognizing dynamic hand gestures suitable for HCI applications.

TH-1870_02610203

Contents

Acknowledgements ii

Abstract iv

1 Introduction 1

1.1 Human Computer Interaction (HCI) . . . 2
1.2 Gesture for Human Computer Interaction . . . 2
1.2.1 Hand gestures . . . 3
1.2.2 Hand gesture modelling . . . 7
1.3 Hand Gesture Recognition for HCI . . . 10
1.3.1 Pattern recognition . . . 11
1.3.2 Gesture recognition as a pattern recognition problem . . . 15
1.3.3 Gesture recognition using data gloves . . . 17
1.3.4 Perceptual interface (PI) . . . 19
1.3.5 Vision based gesture recognition . . . 21
1.3.6 Advantages of vision based gesture recognition . . . 23
1.4 Difficulties in Hand Gesture Recognition . . . 24
1.4.1 Difficulties due to nature of the hand . . . 24
1.4.2 Difficulties due to gesture parameter variance . . . 25
1.4.3 Difficulties in continuous gesture recognition . . . 26
1.4.4 Difficulties associated with image processing techniques . . . 26
1.5 Applications of Hand Gestures . . . 27
1.5.1 Application in human computer interaction . . . 27
1.5.2 Virtual reality . . . 28
1.5.3 Telepresence . . . 28
1.5.4 Automatic sign language recognition . . . 29
1.6 Hand Gesture Recognition: The State of the Art . . . 30
1.6.1 Capturing the hand motion . . . 31

TH-1870_02610203

1.6.2 Static hand posture recognition . . . 35
1.6.3 Dynamic gesture recognition . . . 36
1.6.4 Sign language recognition . . . 43
1.7 Motivation of the Thesis . . . 45
1.8 Objective of the Thesis . . . 47
1.9 Organization of the Thesis . . . 48

2 Recognition of Hand Gestures having Local Motions only 49
2.1 Proposed Scheme for Gesture Recognition . . . 50
2.2 Video Object Plane Generation . . . 53
2.2.1 Overview of MPEG - 4 video coding standard . . . 53
2.2.2 Edge change detection for VOP generation . . . 54
2.2.3 Hausdorff tracker and VOP extraction . . . 57
2.2.4 Method for VOP generation in the proposed gesture recognition scheme . . . 62
2.3 Key VOP Identification and Extraction . . . 68
2.3.1 Proposed method for key VOP selection . . . 70
2.3.2 Key VOP selection using ART shape description . . . 71
2.4 Proposed Finite State Machine Representation of Dynamic Hand Gestures 73
2.4.1 Characteristics of hand gestures . . . 74
2.4.2 Proposed finite state machine (FSM) for gesture representation . . . 75
2.4.3 Training – forming prototype FSMs for gestures in the vocabulary . 75
2.5 Gesture Recognition by FSM Matching . . . 78
2.6 Experimental Results . . . 84
2.6.1 Test gesture sequences . . . 84
2.6.2 Experimental conditions . . . 87
2.6.3 Gesture summarization . . . 87
2.6.4 Gesture classification results . . . 89
2.7 Concluding Remarks . . . 96

3 Recognition of Hand Gestures having Global Motions only 97
3.1 Proposed Scheme for Gesture Recognition . . . 98
3.2 Proposed Trajectory Estimation Algorithm . . . 102
3.2.1 Hand tracking and estimation of the motion vector . . . 102
3.2.2 Determination of the hand image centroid . . . 103
3.2.3 Trajectory formation and smoothing of the final trajectory . . . 105
3.2.4 Key VOP based trajectory estimation . . . 106

TH-1870_02610203

3.2.5 Template trajectory generation . . . 107
3.2.6 Key trajectory point selection on the template trajectory . . . 107
3.3 Feature Extraction from Template Trajectories . . . 108
3.3.1 Static features . . . 109
3.3.2 Dynamic features . . . 112
3.3.3 Forming prototype feature vectors and knowledge-base for gesture matching . . . 114
3.4 Proposed Method for Gesture Classification . . . 115
3.4.1 Gesture trajectory estimation . . . 115
3.4.2 Shape based trajectory matching . . . 115
3.4.3 Nonlinear time alignment and normalization of motion trajectories by dynamic time warping . . . 116
3.4.4 Feature extraction and classification . . . 121
3.5 Experimental Results . . . 122
3.6 Concluding Remarks . . . 129

4 Recognition of Hand Gestures having Local and/or Global Motions 130
4.1 Introduction . . . 131
4.2 Proposed Integrated Gesture Recognition Scheme . . . 133
4.2.1 Gesture type detection . . . 134
4.2.2 Gesture recognition modules . . . 136
4.3 Experimental Details . . . 139
4.3.1 Experimental data sets . . . 139
4.3.2 Experimental setup . . . 140
4.3.3 Experimental results . . . 147
4.4 Concluding Remarks . . . 151

5 Recognition of Continuous Dynamic Hand Gestures 155
5.1 Continuous Gesture Recognition . . . 156
5.1.1 Problems in continuous gesture recognition . . . 156
5.1.2 Gesture spotting methods . . . 158
5.2 Proposed Gesture Spotting Methods . . . 160
5.2.1 Gestural cues for continuous gesture segmentation . . . 160
5.2.2 Gesture boundary detection . . . 162
5.2.3 Movement epenthesis detection . . . 166
5.3 Proposed Algorithm for Gesture Recognition . . . 172
5.3.1 Types of continuous gesture sequences . . . 172

TH-1870_02610203

5.3.2 Gesture type detection 173
5.3.3 Recognition in Type I and Type II gestures 173
5.3.4 Recognition in Type III and Type IV gestures 174
5.4 Experimental Results 174
5.5 Concluding Remarks 182

6 Conclusions **186**
6.1 Summary 186
6.2 Possible Extensions 190

Appendix A **192**
A.1 Distance Transformation Algorithm 192

Appendix B **195**
B.1 MPEG-7 Motion Trajectory Representation 195
B.2 Euclidean and Mahalanobis Distances 197

Publication **198**
C.1 Research papers published/accepted related to this thesis work 198
C.2 Research papers communicated 200

Bibliography **217**

TH-1870_02610203

List of Figures

1.1 Kendon's gesture continuum. 5
1.2 A taxonomy of hand gestures for HCI [Pavlovic *et al.* (1997)]. 6
1.3 Gesture as a stochastic process [Pavlovic *et al.* (1997)]. 8
1.4 Spatial models of gestures [Pavlovic *et al.* (1997)]. 9
1.5 Different hand models. (a) 3D Textured volumetric model. (b) 3D wire-frame volumetric model. (c) 3D skeletal model. (d) Binary silhouette. (e) Contour [Pavlovic *et al.* (1997)]. 10
1.6 General pattern recognition problem. 12
1.7 Different approaches for object recognition. 15
1.8 Hand gesture recognition system. 16
1.9 Cyber gloves (Source: Virtual Technologies, Inc.). 18
1.10 Information flow in perceptual interfaces [Turk & Kolsch (2004)]. 20
1.11 Elements of vision based interfaces (VBI). 21
1.12 Vision based hand gesture recognition. 22
1.13 Variation of hand shapes. 25

2.1 A gesture video sequence showing only local hand motions. 50
2.2 A finger spelling gesture sequence representing "ABCD" in ASL. 51
2.3 Block diagram illustrating the process of VOP extraction. 56
2.4 Computation of Hausdorff distance between two sets of points. 58
2.5 The basic two steps in a Hausdorff tracker. 60
2.6 Flowchart depicting the process of model initialization in a Hausdorff tracker. 61
2.7 Flowchart depicting the process of model update in a Hausdorff tracker. . . 61
2.8 Block diagram for the VOP generation method used in our algorithm. . . . 63

TH-1870_02610203

2.9 Initial model generation, (a)-(b) First two frames (c) Edge image of the first frame (d) Difference image (e) Thresholded difference image (f) Median filtering of threshold difference image (g) Thinning (h) Connected component labelling. The edge image of the first frame is included to show the background edges. 65
2.10 Chamfer 5-7-11 mask. 66
2.11 Original edge image and the corresponding distance transformed image. . . 66
2.12 Results of logical OR and AND operations between horizontal and vertical VOP candidates. (a) Updated model (b) Horizontal candidate (c) Vertical candidate (d) Result of OR operation, where fingers of the hand are lost (e) Result of AND operation, where fingers of the hand are retained. 68
2.13 Extracted VOPs from the hand video sequences 69
2.14 Block diagram for the proposed scheme of gesture recognition. 74
2.15 Summarization of long gesture video sequence. 76
2.16 Finite state machine for gesture representation 76
2.17 Recognition in case of FSMs having different number of states. Dotted arrows indicate the states that are compared to each other. Case I is an ideal case where the number of states in both input and database FSMs are same. Case II shows an input with one missing state while Case III shows an input with one redundant state. 80
2.18 Variation of threshold flexibility for recognition. 82
2.19 Indexing of FSMs by ART shape descriptions, where $(n \geq m)$. 83
2.20 Four frames of the experimental dynamic gesture sequences (a) Close to open (b) Open to close (c) Close to one and (d) Close to two, respectively. . 86
2.21 Key VOP selection for "Rotate" sequence 90
2.22 Key VOP selection for "No" sequence 90
2.23 Key VOP selection for "Clic" sequence 91
2.24 Key VOP selection for "Stop Grasp OK" sequence 91
2.25 Some of the experimental static hand shapes showing "Close", "Open" "One" and "Two", respectively. 93

3.1 Gestures representing "two", "three", "wavy hand" and "square" respectively. 98
3.2 2D hand motion trajectory in spatio-temporal space. 99
3.3 Centroids calculated by motion vector and moment equations in VOPs. . . . 105
3.4 Pictorial illustration for estimation of centroids in VOPs. 105
3.5 Distances of the key trajectory points from the center of gravity. 110
3.6 Typical velocity plots of gestures representing (a) Circle and (b) Square. . . 114

TH-1870_02610203

3.7 Trajectory shape matching. 116
3.8 Correspondence between points in two curves in DTW. 118
3.9 Finding the optimal warping function using DTW. 119
3.10 Key point selection in the input trajectory by DTW matching. 120
3.11 Training algorithm. 123
3.12 Classification algorithm. 124
3.13 Hand gestures showing different motion trajectories corresponding to "One", "Two", "Three", "Five", "Six", "Seven", "Alpha", "Circle", "Square" and "Diamond" respectively. 126
3.14 Original extracted trajectories are shown in first row and smoothed trajectories for the same are shown in second row of the figure. The gestures represent (a) Circle (b) Square (c) One (d) Two (e) Seven and (f) Six, respectively. 127
3.15 Trajectories of (a) Circle (b) Square (c) One (d) Two (e) Seven and (f) Seven, respectively extracted from key VOPs. 127

4.1 Hand moves in a trajectory showing different hand shapes. 131
4.2 Recognition of gestures having both local and global motions. 132
4.3 Block diagram for the proposed gesture recognition system. 135
4.4 Movement of VOP centroids in 2D plane in case of gestures having local motion only. 137
4.5 Movement of VOP centroids in 2D plane in case of gestures having global motion. 137
4.6 Cluster of centroids in 2D plane in case of gestures having local motion only. 138
4.7 Mapping of key points from template to test trajectory. 140
4.8 Gestures having both local and global hand motions. 142
4.9 Some static hand postures of the native Indian sign language. 143
4.10 Dynamic hand signs (pictographs) of the native Indian sign language taken from William Tomkin's online database. 145
4.11 Five frames of the signs from the native Indian sign language (a) Arise (b) Fire (c) Yes (d) Give and (e) Jump, respectively. 146
4.12 Test setup and command window for the proposed gesture recognition system 148
4.13 Cluster of centroids in 2D plane for all gestures considered in Chapter 2. . . 149
4.14 Movement of centroids in 2D plane for the "Square" gesture considered in Chapter 3. 149

5.1 A hierarchy of gesture components [Eisenstein & Davis (2005)]. 162

TH-1870_02610203

5.2 VOP model for a portion of a continuous gesture sequence showing movement epenthesis. 163
5.3 Gesture sequence starting point detection 164
5.4 Basic hand strokes that are used in building different gesture trajectories. . . 165
5.5 (a) Single hand stroke for a gesture (b) Multiple hand strokes for a gesture (c) Multiple strokes for a sentence of sign language. 165
5.6 Example of a typical velocity plot for two sequentially connected gestures having global motion in a continuous video stream. 168
5.7 Fuzzy membership functions defining different ranges of speed. 170
5.8 Fuzzy membership functions defining different ranges of change in speed. . 170
5.9 Examples of gestures having intermediate movement epenthesis 173
5.10 Static hand poses 'One', 'Two', 'IV hand', 'Cartridge' and 'Friend' respectively used in our 1st experiment. 175
5.11 A portion of a continuous gesture sequence showing transition from one gesture to the next used in our 2nd experiment. 178
5.12 Trajectories of the gestures used in our experiments: (a) Gestures that produce movement epenthesis, used in our 3rd set of experiments, (b) Gestures that do not produce movement epenthesis, used in our 4th set of experiments. 179
5.13 (a) A continuous motion trajectory without movement epenthesis (b) A continuous motion trajectory with movement epenthesis (c) Segmented out trajectories. 180

A.1 Chamfer masks usedfor computing distance transformation. 193
A.2 Edge image and corresponding distance transformed image. 194
B.1 Example of motion trajectory representation (One-dimensional) and corresponding bitstream structure. 196

TH-1870_02610203

List of Tables

2.1 Different parameters used for VOP generation 88
2.2 Table showing summarization of the gesture sequences 92
2.3 Experimental Results: Confusion matrix showing the gesture recognition rate using the proposed FSM-based method. 93
2.4 Experimental Results: Confusion matrix showing the gesture recognition rate using HMM classifier. 94
2.5 Table showing static gestures recognition rates 95

3.1 Gesture patterns for training and testing. 126
3.2 Experimental results demonstrating the percentage of accuracy in gesture recognition using our proposed method. 128

4.1 Experimental Results: Confusion matrix showing the gesture recognition rate using the proposed gesture recognition scheme. 152
4.2 Recognition rate for static hand signs taken from the native Indian sign language. 153
4.3 Recognition rate for isolated words in the native Indian sign language. . . . 154

5.1 Gesture motion phases and corresponding motion parameters 171
5.2 Gesture recognition rates for Type I gesture sequence. 176
5.3 Gesture recognition rates for Type III (3^{rd} set of experiments) gesture sequence. 181
5.4 Average acceleration in cm/sec^2 in continuous gesturing 182
5.5 Gesture recognition rates for Type III (4^{th} set of experiments) gesture sequence. 183
5.6 Gesture recognition rates for Type IV gesture sequence. 184
5.7 Average gesture recognition rates for set of experiments. 185

6.1 Average Gesture Recognition Rates 189

TH-1870_02610203

Chapter 1

Introduction

Gesture recognition is an important area of research in the field of computer vision. The most recent application of computer vision is the development of the Human Computer Interactive (HCI) systems. In the framework of human-machine nonverbal communication, the visual interface becomes important in establishing communication channels for understanding human intention from the human behaviour, such as the facial expressions, hand gestures, etc. With the increased interest in HCI, there has been rapid growth of studies related to gesture recognition in recent years. Hand gesture recognition from visual images forms an important part of this interface. It has potential applications in machine vision, virtual reality, robotic control and so on.

The goal of this research work is to build a system that recognizes natural hand gestures in a vision-based setup. The work aims at recognizing wide classes of isolated hand gestures having different spatio-temporal characteristics. Simultaneously, we also try to recognize continuous stream of hand gestures by detecting gesture boundaries.

TH-1870_02610203

1.1 Human Computer Interaction (HCI)

Human-computer interaction (HCI) is the study of people, computer technology, and the ways they influence each other. It involves the design, evaluation, and implementation of interactive computing systems for use by a human. In general, HCI is a very broad interdisciplinary field involving computer scientists, psychologists, cognitive scientists, and other disciplines [Donato *et al.* (1999) and Sebe *et al.* (2004)].

The function of an HCI system is to enable a human to communicate with computer in a way which is more natural and convenient than the traditional input devices such as a mouse, a keyboard, a joystick, etc. Accordingly, the primary aim of the HCI research is to build up a platform through which a human can communicate with a computer using "natural" means that humans employ to communicate with one another [Pavlovic *et al.* (1997)]. With this motivation, automatic speech recognition had been a topic of research for decades, and several commercially successful speech-based HCI systems have been developed [Georgila *et al.* (2003)]. Inspired by the success with speech-based HCI, there has been a growing interest in trying to introduce other human-to-human communication modalities into HCI in recent years. These include gestures formed by the movement of one or more body parts. Human gesture is one common and natural means of non-verbal communication among people. Next to speech, it is the most obvious choice for natural interfacing between a human and a computer. A gesture-based HCI system allows a person to input commands using natural movements of the hand, head and other body parts. It has applications in industry such as to control a robotic arm. It also allows a person to communicate with a computer via a sign language and thus allows people with disability to interact with a computer more easily. In recent years, there has been a significant effort devoted to body motion analysis and gesture recognition.

1.2 Gesture for Human Computer Interaction

A gesture is an expression made by some bodily movement and is used by a person to express some meaning. It is generally made up of facial expression, hand expression and/or movement of other body parts. In general, the most common form of gestures is made up

TH-1870_02610203

of movements of the hand and/or arm associated with facial expressions. In this, the hand is used to make different message signs while facial movements are used to reflect the mood and emotion of the person.

According to Knapp and Hall "*Gestures are movements of the body (or some part of it) used to communicate an idea, an intention, or a feeling. Many of these activities are made with the arms/hands. Gestures perform many functions. They may replace speech, regulate the flow and rhythm of interaction, maintain attention, add emphasis and/or clarity to the speech...*" [Knapp & Hall (1997)]. The Webster's Dictionary defines gestures as "*...the use of motions of the limbs or body as a means of expression; a movement usually of the body or limbs that expresses or emphasizes an idea, sentiment, or attitude*". Therefore, the term gesture refers to some semantic interpretation that is associated with a particular movement of the body. In particular, hand and arm gestures had been covered extensively in subjects like social science and conversational and behavioral psychology [Efron (1941), McNeill (1992), Quek *et al.* (2000)]. However, in the context of HCI, the notion of gestures is somewhat different. In an HCI system, hand and other body parts are used as natural manipulators as well as for conveying messages for control of computers and machine functions [Pavlovic *et al.* (1997)].

1.2.1 Hand gestures

A hand gesture is a means of communication, similar to the spoken language. Hands are our most dextrous body parts, and they are used in both manipulation and communication. The hand is the most fluent and articulate part of the body, capable of expressing almost infinite meanings. Hand gestures play the role of a natural and intuitive communication mode for all human dialogues and constitute the majority of gestures used by humans. They range from simple actions like using the hand to point at an object to the more complex ones that express feelings and allow one to communicate with others.

Hand gestures are used widely in every day life, from being used to emphasize speech in ordinary conversation, to being used as a vital communication tool for deaf and mute people. When hand gestures are used to communicate words for the deaf, each individual gesture is an element of a large set of gestures called the *sign language*. There are many

TH-1870_02610203

variations of sign languages in use; they have been invented to meet the communicative needs of deaf communities in various cultures. When used in association with speech, the meanings of different gestures also depend on the people and the culture. That means, there is no universal set of gestures used by humans all over the world. There may be a gesture having different meanings among people of different countries, races and cultures. This creates ambiguity and makes it difficult to model hand gestures for general use. Nevertheless, a set of specific hand gestures can be always predefined in an application so that the ambiguity can be limited.

Gesture continuum

Kendon described a *gesture continuum* as depicted in Figure 1.1. In his definition, there are five different kinds of gestures. As the list progresses from left to right in Figure 1.1, the association of the gesture with speech declines, language properties increase, spontaneity decreases, and social regulation increases [Kendon (1988) and Kendon (1990)]. Following are the five kinds of gestures:

- **Gesticulation:** These are spontaneous and natural movements of the hands that accompany speech. They themselves have no meaning and are used only to emphasize some spoken words or phrases.
- **Language-like gestures:** These are meaningful gestures that are used as replacement of some particular words or phrases during conversation.
- **Pantomimes:** These are gestures that define objects or actions. They may occur with or without accompanying speech.
- **Emblems:** These are familiar gestures such as "V for victory", "Thumbs-Up" etc. used commonly by people and bear some special meanings. They are often culturally specific.
- **Sign languages:** These are gestures used mainly by the deaf and mute community as total replacement of speech. These gestures form a linguistic system with proper

TH-1870_02610203

grammar and well defined syntax. Examples of sign languages are the American Sign Language (ASL), the British Sign Language, the native Indian Sign Language etc.

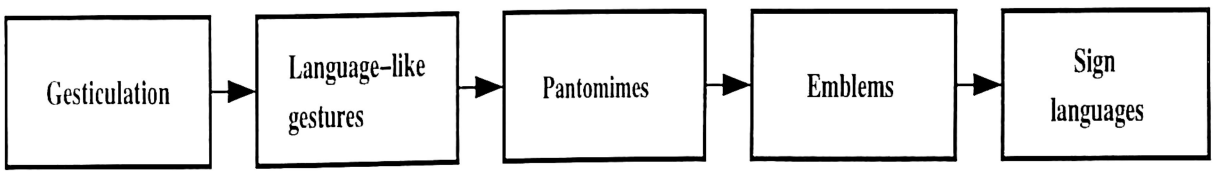


Figure 1.1: Kendon's gesture continuum.

Types of hand gesture

The richness and complexity in gestural communication makes the problem of gesture analysis and gesture study a difficult task. Despite this, researchers have been able to understand and describe the nature of gesture to a significant degree. For example, Stokoe represented gestures in terms of four aspects comprising the shape, the position, the orientation and the movement of the hand [Stokoe (1960)]. Based on this, hand gestures are described in terms of the following four major attributes:

1. Hand configuration (*i.e.*, posture),
2. Palm orientation,
3. Hand position, and
4. Hand movement.

In general, hand gestures can be classified into two categories: (1) *Static hand gestures* which rely only on the information about the flexure angles of the fingers, and (2) *dynamic hand gestures* which rely not only on the flexure angles but also on the hand trajectories and orientations. Dynamic hand gestures can be further divided into two subclasses. The first subclass is the gesture with *local hand motion* and consists of hand gestures involving movement of the fingers only without changing the spatial position of the hand. The second subclass is the hand gesture with *global hand motion* and consists of hand gestures that involve movement of the hand/arm in space. Therefore, dynamic gestures are composed

TH-1870_02610203

of at least two different hand shapes and/or positions connected sequentially in time. Samples of dynamic hand gestures are spatio-temporal patterns. A *static hand gesture* may be viewed as a special case of dynamic gesture with no temporal variation of the hand shape and position.

Taxonomy of hand gestures

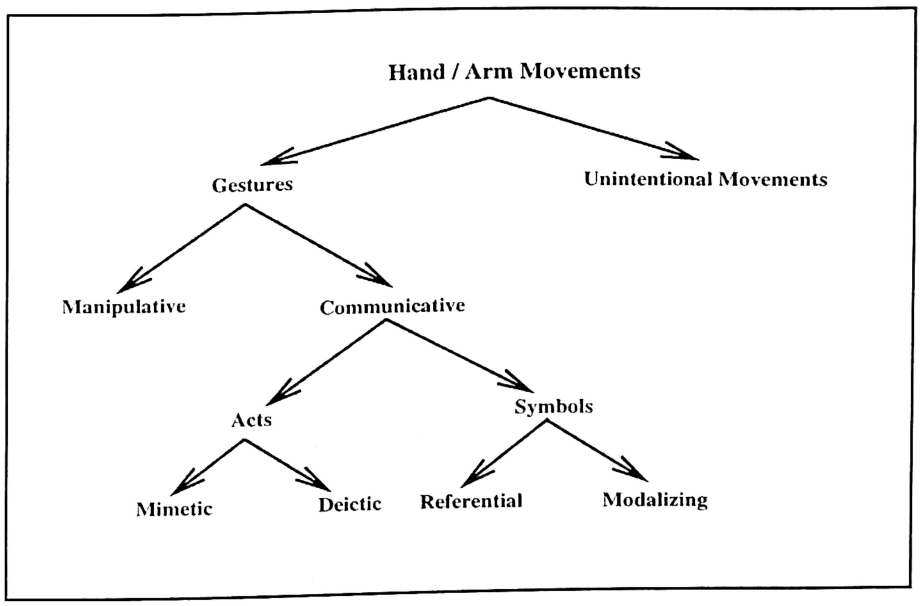


Figure 1.2: A taxonomy of hand gestures for HCI [Pavlovic *et al.* (1997)].

As shown in Figure 1.2, all hand/arm movements can be classified as either meaningful gestures or unintentional movements. Unintentional movements are those movements that do not convey any meaningful information. On the other hand, gestures generally convey some meaning and may be *manipulative* or *communicative*. Manipulative gestures are the ones that are used to make objects in an environment to act as per the gesture commands (object movement, rotation, etc). They can serve as a natural way to interact with virtual objects (virtual reality). Communicative gestures are those that have an inherent purpose of conveying some messages. They are usually accompanied by speech. Communicative gestures can be either acts or symbols. Symbols are those gestures that have a linguistic role. A sign language is an important example of communicative gestures consisting of symbols

TH-1870_02610203

or signs. In the HCI context, communicative gestures are the most commonly used gestures since they can often be represented by different static hand postures. A symbol may represent some referential action or be used as a modalizer. Finally, acts are gestures that are directly related to the interpretation of the movement itself. Such movements are classified as either *mimetic* (which imitate some actions) or *deictic* (pointing acts). In mimetic gestures, motions form an object's main representative feature.

1.2.2 Hand gesture modelling

The first step in a hand gesture recognition task is to choose a mathematical description or model of the gesture. In fact, the scope of a gestural interface for HCI is directly related to the proper modelling of hand gestures. Modelling of hand gestures depends on the intended application within the HCI context. For some applications, a very coarse and simple model may be sufficient. However, if the purpose is a natural-like interaction, a model has to allow many, if not all, natural gestures to be interpreted by the computer.

A gesture model should consider both the spatial and temporal characteristics of the hand and its movements. The models are generally represented in parametric forms, with the parameters belonging to their respective parameter spaces [Pavlovic *et al.* (1997)]. Based on this concept, a hand gesture may be defined as a stochastic process in the gesture model parameter space and over a suitably defined time interval. Each gesture can be represented as a trajectory in the model parameter space. Figure 1.3 shows two different gestures in a three-dimensional parameter space. Because of the stochastic property in the definition of gestures, no two samples of the same gesture will result in exactly the same hand and arm motions or the same set of visual images *i.e.,* gestures suffer from spatio-temporal variation.

Temporal modelling of gestures

The temporal progression of hand gestures, especially those that accompany speech generally consists of three stages: *pre-stroke*, *stroke*, and *post-stroke* [McNeill (1992)]. The pre-stroke prepares the movement of the hand. The hand waits in this ready state until the speech arrives at a point of time when the stroke is to be delivered. The stroke is the

TH-1870_02610203

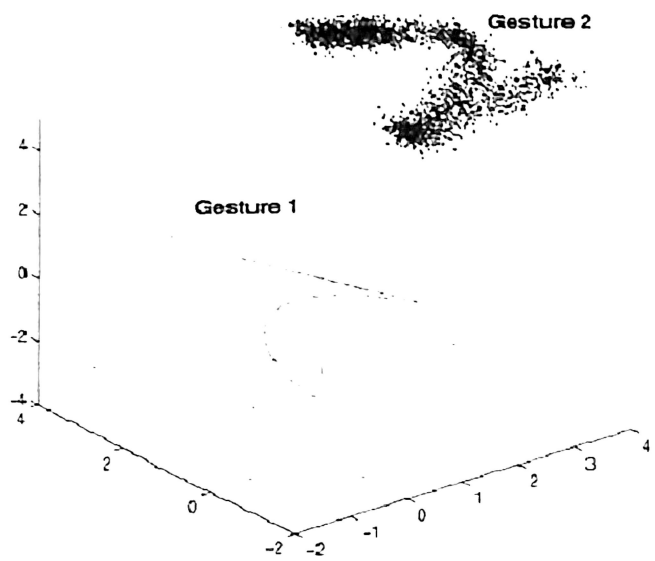


Figure 1.3: Gesture as a stochastic process [Pavlovic *et al.* (1997)].

actual gesturing phase when the hand moves very fast. Thus, the stroke is often characterized by a peak in the hand's velocity. The hand is retracted during the post-stroke. This phase is referred to as *co-articulation* and is strongly influenced by the gesture that follows. Co-articulation is the phenomenon in which one gesture influences the next in a temporal sequence [Vogler & Mextaxas (1997), Shamaie *et al.* (2001)].

Spatial modelling of gestures

Dynamic hand gestures are generally seen as hand and arm movements in the 3D space. As illustrated in Figure 1.4, there are two approaches to gesture modelling that may be adopted in hand gesture analysis and recognition. The first approach tries to identify gestures directly from the visual images and is called the *appearance-based* modelling. This means that the model parameters are not derived from the 3D spatial description of the hand but are obtained by relating the appearance of a given gesture to the appearances of a set of predefined template gestures. Parameters of such models may be either the images themselves or some features derived from the images. Appearance-based models directly link the appearance of the hand and arm movements in visual images to specific gestures. A different kind of appearance-based modelling uses 2D hand image sequences as ges-

TH-1870_02610203

ture templates. Here each gesture in the gesture vocabulary is modelled by a sequence of temporally connected representative images, each image corresponding to one view of the gesturing hand [Pavlovic *et al.* (1997)]. For instance, a video of the entire human hand in motion may be used as the template for each of the gestures contained in the gesture vocabulary [Darrell *et al.* (1996), Darrell & Pentland (1995)]. Contrary to this, images of fingers only may be employed as templates in a finger tracking application [Crowley *et al.* (1995)].

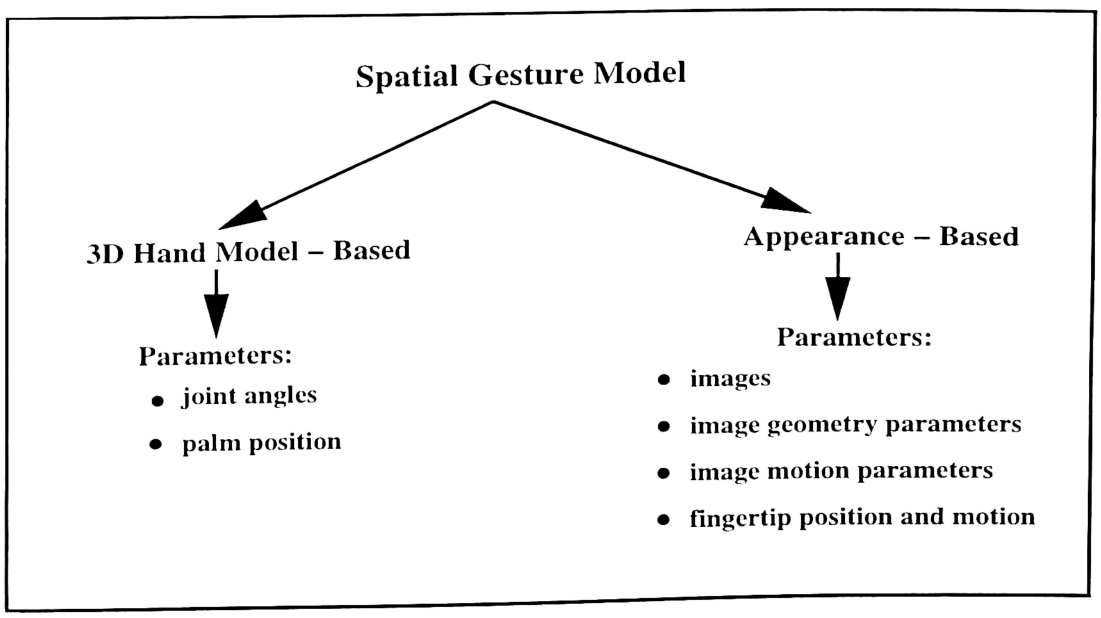


Figure 1.4: Spatial models of gestures [Pavlovic *et al.* (1997)].

The second approach to gesture modelling is based on *3D hand model*. This description is derived using articulated model of the human hand and arm to estimate their motion parameters. Such movements are later recognized as gestures. The models obtained in this approach may be classified in two board categories:

- volumetric models, and
- skeletal models.

Volumetric models are used to describe the 3D visual appearance of the hand and arms. They are generally employed in computer animation, but have also been used in computer vision applications [Thalmann & Thalman (1990)]. In the field of computer vision, volu-

TH-1870_02610203

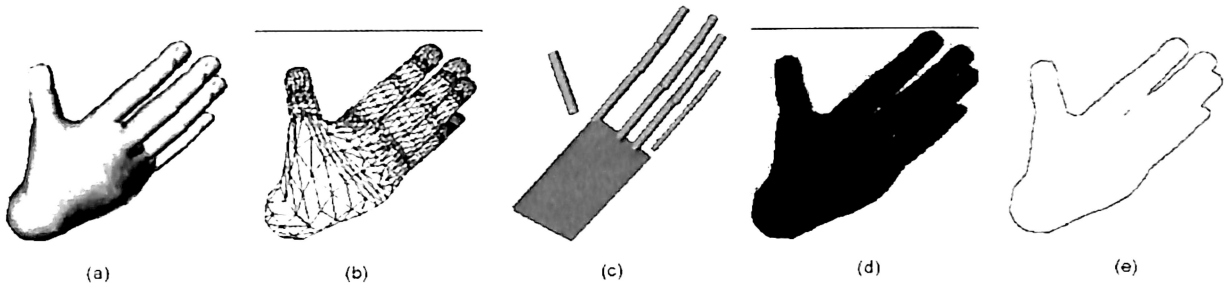


Figure 1.5: Different hand models. (a) 3D Textured volumetric model. (b) 3D wireframe volumetric model. (c) 3D skeletal model. (d) Binary silhouette. (e) Contour [Pavlovic *et al.* (1997)].

metric models of the human body are used for analysis-by-synthesis tracking and recognition of the body's posture [Koch (1993), Wren *et al.* (1996)]. Volumetric models require more parameters than image-based models and hence lead to more expensive computation during the matching process. A more appealing approach, suitable for real-time computer vision, is based on modelling of the human body using simple 3D geometric structures [Rourke & Badler (1980)]. In this approach, structures like cylinders, spheres, ellipsoids and generalized cylinders are used to approximate the shape of simple body parts, like finger links, forearm, or upperarm [Azarbayejani *et al.* (1996), Clergue *et al.* (1995), Downton & Drouet (1991), Etoh *et al.* (1991) and Gavrila & Davis (1995)]. Instead of dealing with all the parameters of a volumetric hand and arm model, models with a reduced set of equivalent joint angle parameters together with segment lengths are often used. Such models are known as *skeletal* models. Skeletal models are extensively studied in human hand morphology and biomechanics [Thompson (1981)].

1.3 Hand Gesture Recognition for HCI

As discussed earlier, gestures may be described as the manner in which a person moves his body and limbs to express an idea or a sentiment. Gestures are natural means of conveying

TH-1870_02610203

information and used for everything from pointing at a person to get his/her attention to conveying information in day-to-day life. This has motivated the use of gestures for communicating with computers. They may provide an attractive and user-friendly alternative to interface devices like keyboards, mice and joysticks. Therefore, a very interesting current topic in technology development is to create a system which can identify/interprete human gestures automatically and use them to convey information or for device control.

To exploit the use of gestures in HCI, it is necessary to equip the computer with the algorithm by which the gestures can be interpreted. The first step towards interpretation of gestures requires that dynamic and/or static configurations of the human hand, arm, and even other parts of the human body be measurable by the machine. Initial attempts to solve this problem resulted in mechanical devices that directly measure hand and/or arm joint angles and their spatial positions. One such commonly used device is the *DataGlove* that accomplishes this goal, but has an unnatural feeling and is cumbersome to the user. The second approach is the *vision-based* approach that determines hand configuration information by analyzing the gesture image/video captured by one or more cameras. We will discuss these two approaches in the sections to follow. Irrespective of the method for acquisition of gesture data, gesture interpretation involves recognition of patterns. We also present a brief outline of the concept behind pattern recognition.

1.3.1 Pattern recognition

Pattern recognition (PR) is an important field in computer science. It is concerned with the description and/or classification (recognition) of a set of patterns. Patterns may be concrete patterns, such as characters, pictures, music, physical objects, etc., or may be abstract patterns that include argument, solution to a problem, etc. Recognizing concrete patterns necessitates sensory (visual and acoustic) recognition while abstract patterns need conceptual recognition.

Human beings perform the task of recognizing objects around them in almost every instant of time and act in relation to these objects. When a person perceives a pattern, he makes an inductive inference and associates this perception with some general concepts he has derived from his past experience. Research in pattern recognition is concerned with

TH-1870_02610203

building human-like intelligent systems that can perform the same task without any human intervention. The advantage of such a system is that it is possible to do bulk recognition in less time compared to that by a human. But, it generally lacks the capability to handle any ambiguity and/or uncertainty which a human recognition system can do quite efficiently. However, with the advances in technology, it has become possible to build recognition systems with comparable reliability.

As illustrated in Figure 1.6, a pattern recognition problem can in general be split into two subproblems, *viz.*, (1) Problem of representation (2) Problem of decision.

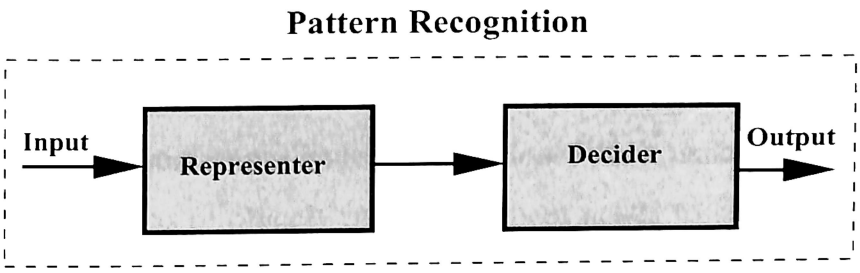


Figure 1.6: General pattern recognition problem.

The problem of representation is concerned with converting input data (*e.g.*, an image) to an output representation, often in the form of a set of features extracted from the input data. The representation is in a form that is suitable for the decision process to recognize the presented pattern accurately.

Steps in pattern recognition

The two main tasks involved in any PR system are *feature extraction/selection* and *classification* [Duda *et al.* (2002)]. Feature extraction/selection is the process of choosing a set of attributes from a given pattern that best represents the pattern. Classification assigns input data into one or more of all the pre-specified pattern classes. Accordingly, we may now have the following definitions.

Pattern class: This is a set of patterns, sharing some common attributes, known to originate from the same source. The key in most PR applications is to identify suitable

TH-1870_02610203

attributes (features), to define a good measure of similarity and formulate an appropriate matching process.

Classification: This is the process of assigning input object into one or more of $\mathcal{K}$ classes via extraction and analysis of significant attributes present in the object. Classification of objects is an important area of research and has practical applications in a variety of fields, including pattern recognition and vision analysis.

Recognition: Recognition (or identification) involves comparing an input image to a set of models in a database. A recognition scheme usually determines confidence scores or probabilities that define how closely the image data fits into each model.

Detection: Detection is sometimes called recognition, which makes sense if there are distinct different classes of objects and one of them needs to be recognized. A special case of recognition is verification or authentication, which judges whether an input data belongs to one particular identity or not. An important application of verification is in biometrics, which has been applied to faces, fingerprints, gait, etc. [Choudhury *et al.* (1999)].

Types of classifier

1. **Crisp classifier:** In this, an input pattern is classified to only one of the $\mathcal{K}$ classes and is excluded from belonging to the remaining $(\mathcal{K} - 1)$ classes. That means, the classifier takes crisp (hard) decision for classification.

2. **Fuzzy classifier:** A fuzzy classifier assigns an input pattern to each of the $\mathcal{K}$ classes with different degrees of belongingness, generally measured using *fuzzy membership functions*. This is soft classification.

3. **Classifier with reject option:** Computer recognition may not be appropriate for decision making in cases where there exists high degree of uncertainty, imprecision and/or ambiguity. In order to avoid making of any wrong decision, rejection of such an ambiguous case is generally recommended. An input pattern is rejected by a classifier when there is no clear 'winner' from among the $\mathcal{K}$ pattern classes.

TH-1870_02610203

Training

Training is defined as the process of developing the decision rules in a classifier from prior knowledge about the problem at hand, via extraction of significant attributes of each pattern in a given set of training patterns that bears all relevant prior information about the problem. In other words, it is the process of finding an appropriate classification function f_C that maps a given input point in the pattern space to one of the classes in the decision space to which the input pattern may belong. The process involves learning characterizing properties for each class from the set of typical training patterns that provides significant information on how to associate input data with output decisions. Training may be either supervised or unsupervised, depending on whether the class labels of the training patterns are known or not.

Testing

Once a classifier is designed via training, it is necessary to check the performance of the designed system. This is called testing. Testing means submitting the test patterns to the designed classifier and then evaluating the performance of the system. Generally, the classification accuracy of the system is taken as the measure for its performance that is defined as

$$Accuracy\ Rate = \frac{Total\ number\ of\ correctly\ recognized\ test\ patterns}{Total\ number\ of\ test\ patterns} \tag{1.1}$$

Accordingly, the classification error rate is given as

$$Error\ Rate = 1 - Accuracy\ Rate \tag{1.2}$$

In a classifier with reject option, the accuracy rate is given by the same expression as in equation (1.1), whereas the error rate and the rejection rate are given as

$$Error\ Rate = \frac{Total\ number\ of\ incorrectly\ recognized\ test\ patterns}{Total\ number\ of\ test\ patterns} \tag{1.3}$$

$$Rejection\ Rate = \frac{Total\ number\ of\ rejected\ test\ patterns}{Total\ number\ of\ test\ patterns} \tag{1.4}$$

TH-1870_02610203

Here we observe that

$$Error\ Rate + Rejection\ Rate = 1 - Accuracy\ Rate \tag{1.5}$$

1.3.2 Gesture recognition as a pattern recognition problem

Gesture recognition is basically an object recognition problem [Birk & Moeslund (1996)]. Given an object recognition problem, it can be viewed in two different ways following the appearance-based and model-based approaches as discussed earlier. As depicted in Figure 1.7, there is no abstraction level in an appearance-based method while with a model-based method an abstraction level is present, *e.g.*, providing implicit information of the presence of a joint between the palm of the hand and a finger. Clearly, gesture recognition can be viewed as a problem of pattern recognition, where the input from either of the two sensors described in Sections 1.3.3 and 1.3.5 is to be classified. As with any pattern recognition problem, the task of gesture recognition also consists of two major components: feature extraction and classification.

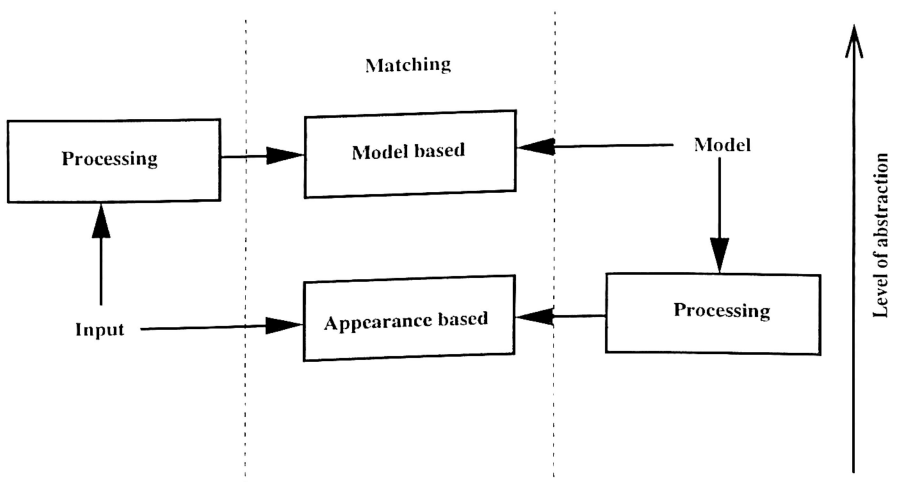


Figure 1.7: Different approaches for object recognition.

As with any pattern recognition problem, the task of gesture recognition also consists of two major components: feature extraction and classification. The design process for most gesture recognition systems may be split into two phases: (1) the training phase and (2)

TH-1870_02610203

the recognition phase. The training phase allows the user to 'tell' the system what a new gesture 'looks' like. This phase captures the spatio-temporal variations among different gesture samples and develops a set of models representing the gestures in the vocabulary. These gesture models allow for spatial and temporal variations among samples of the same gesture. They must also allow for continuous gesture recognition, that is, when the user is allowed to continually move his hand making any number of gestures in sequence and the system must be able to identify meaningful gestures embedded in this continuous hand movement. The trained system then recognize similar gestures when it is deployed.

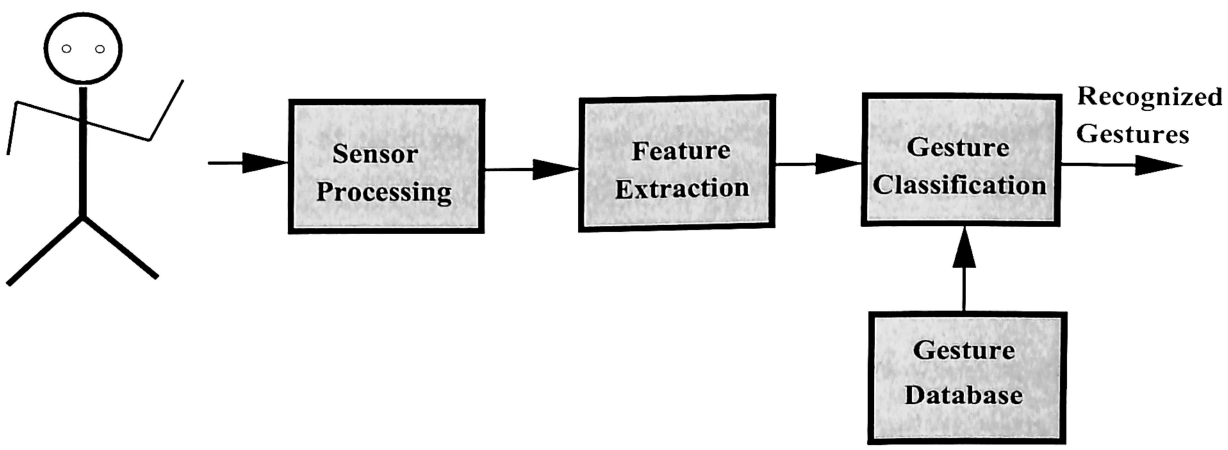


Figure 1.8: Hand gesture recognition system.

As shown in Figure 1.8, the first step in gesture recognition is to sense different aspects of gesture, *viz.*, hand position, hand configuration and hand movement. This may be done directly with sensing devices such as magnetic field trackers, DataGloves and datasuits which are attached on the user, or indirectly using cameras and computer vision techniques. Sensing technologies differ from one another in terms of accuracy, resolution, latency, range of motion, user comfort and cost. Therefore, integration of multiple sensors in gesture recognition is a complex task. In case of gesture recognition using a sensing device, the main interest is primarily in the interpretation of the sensor data to recognize the gestural information. The output of the sensor processing step is a time-varying sequence of parameters describing positions, velocities, and angles of the fingers. The parameters produced by the sensors may be transformed into a global coordinate space, processed to produce

TH-1870_02610203

sensor-independent features, or used directly in the classification step. Because hand gestures are highly variable from one person to another and from one example to another, it is essential to capture the essence of a gesture, its invariant properties and use these to represent the gestures. The choice of representation is a significant issue in building gesture recognition systems. The systems may be fully trained when in use, or they may adapt over time to the current user.

Static hand gesture recognition may be accomplished by a straightforward implementation of the steps in Figure 1.8, using *template matching*, *geometric feature* classification, *neural networks*, or other standard pattern recognition techniques. Dynamic hand gesture recognition, however, requires consideration of temporal events. This is typically accomplished through the use of techniques such as *time-compressing templates*, *dynamic time warping*, *finite state machines*, *hidden Markov models* (HMMs), and *Bayesian networks*. Another important issue in dynamic hand gesture recognition is the *hand tracking* in the gesture video sequence. Tracking may be considered as a repeated frame-by-frame process of detection of objects that locates objects and reports their changes in positions over time [Gavrila (1999)]. Since straightforward tracking is computationally expensive and time consuming, various methods exist for real-time application. These methods improve tracking by explicitly taking temporal continuity into account and using prediction to limit the space of possible solutions.

1.3.3 Gesture recognition using data gloves

In the glove-based approach, some mechanical or optical sensors are usually attached to a glove which transduces finger flexions and abductions into electrical signals in such a way that the hand configuration at every instant of time is easily determined. The relative position of the hand is determined by some additional sensors (magnetic or acoustical) mounted on the glove. A detailed survey of glove-based input devices can be found in [Sturman & Zeltzer (1994)].

Over the years many advanced glove devices have been designed such as the Data Glove, Sayre Glove, Dexterous HandMaster and PowerGlove etc. [Watson (1993)]. The most successful commercially available glove is by far the VPL DataGlove. It was devel-

TH-1870_02610203



Figure 1.9: Cyber gloves (Source: Virtual Technologies, Inc.).

oped by Zimmerman during the 1970s [Zimmerman & Lanier (1987)]. It is based on optical fiber sensors along the back of the fingers. Flexing the fingers bends the optical fibers, attenuating the light they transmit. The analog signal is sent to a processor which determines the joint angles based on the light intensity, as calibrated for a given user. A gesture recognition software package included with the DataGlove makes it possible for the user to define commands for different hand configurations. Another useful sensory device is the Polhemus 3SPACE sensor in which the position and orientation of the hand can be calculated. It can track six degrees of freedom using electromagnetic induction. A transmitter and a fixed receiver mounted on the glove is the basis for computing the position and orientation of the hand. Glove-Talk is another gesture-to-speech interface, developed by Fels and Hinton, that uses a VPL DataGlove connected to a DECtalk speech synthesizer via five independent neural networks [Fels & Hinton (1993)]. They defined a 203 gesture-to-word vocabulary and used Glove-Talk to map complete gestures to complete words. The CyberGlove developed by James Kramer, is a light-weight flexible glove with sensors which accurately measures the positions and movements of the fingers. There are 18 or 22 sensors in the CyberGlove and a serial cable is connected to the host computer. Baudel and Beaudouin-Lafon developed a real-time glove-based Charade system in which hand gestures are used to control browsing in a hypertext presentation [Baudel & Beaudouin-Lafon (1993)]. It can

TH-1870_02610203

recognize sixteen hand gestural commands, each of which comprises three phases: start posture, dynamic phase, and end posture. The commands are distinguishable based on the start posture as well as the dynamic phase. A data glove with reducing sensors based on magnetic induction has been presented in [Fahn & Sun (2005)]. This data glove has the ability of measuring 10 degrees of freedom using only five sensors that are attached on the palm.

Although glove-based gesture recognition is generally very accurate, the main disadvantage is that the gestural interfaces require the user to wear a cumbersome device, and carry a load of cables that connect the device to a computer. This hinders the ease and naturalness with which the user can interact with the computer controlled environment.

1.3.4 Perceptual interface (PI)

In recent years, a new area, referred to as *perceptual interface* (PI), has emerged. The general focus of the researches in this area is to integrate multiple perceptual modalities (such as computer vision and speech processing) into the user interface. In the context of computer vision technology, the primary aim of the PI is to use vision as an effective input modality in human-computer interaction.

PI are characterized by interaction techniques that combine an understanding of natural human capabilities with computer I/O devices and machine perception and reasoning. They seek to make the user interface more natural and compelling by taking advantage of the ways in which people naturally interact with one another. These include both verbal and non-verbal communications. Broadly defined, perceptual interfaces are highly interactive, multimodal interfaces that enable rich, natural, and efficient interaction with computers [Turk & Kolsch (2004) and Turk (2004)]. A multimodal interface is a system that combines two or more input modalities in a coordinated manner. Perceptual interfaces are inherently multimodal.

There are two main features of PIs. First, they are highly interactive. Unlike traditional passive interfaces that wait for users to enter commands before taking any action, they actively sense and perceive the world and take actions based on goals and knowledge at various levels. Second, they are multimodal, making use of multiple perceptual modalities

TH-1870_02610203

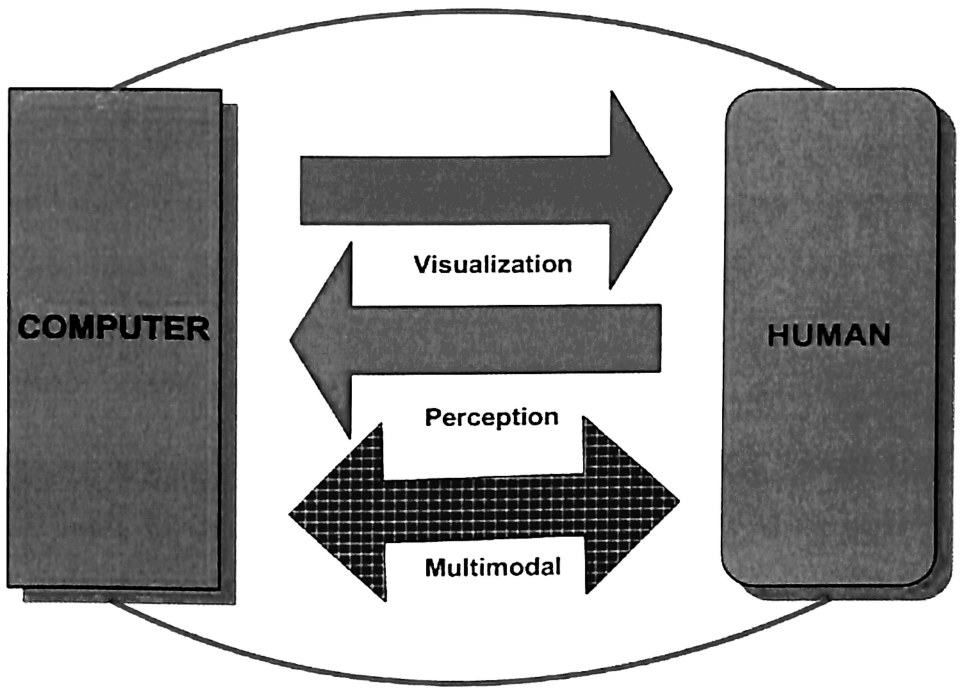


Figure 1.10: Information flow in perceptual interfaces [Turk & Kolsch (2004)].

in both directions – from the computer to the user, and from the user to the computer. They take advantage of a wider range of modalities, either sequentially or in parallel. The concept of perceptual interface is depicted in Figure 1.10.

The motivation behind perceptual interfaces is two-fold: (1) the changing nature of computers, and (2) the desire for a more powerful, compelling user experience than what has been available with graphical user interfaces (GUI). Neither mouse nor keyboard, for example, is very appropriate for communicating 3D information or human emotions. Interfaces must become more natural, intuitive, adaptive, and unobtrusive to accommodate a wider range of scenarios, tasks, users and preferences. These are the primary goals of research in perceptual interface. There has been increased interest in building interfaces that use these technologies to use the natural human capabilities to communicate via speech, gesture, expression, touch, etc. The multimodal context-sensitive human-computer interaction is likely to become the single most widespread research topic of the artificial intelligence research community [Pentland (2000) and Jaimes & Sebe (2005)].

Computer vision is one vital element for perceptual interfaces. Whether alone or in con-

TH-1870_02610203

junction with other perceptual modalities, visual information provides useful and important cues to interaction. The presence, location, and posture of a user may provide important contextual information; a gesture or facial expression may be a key signal; the direction of gaze may disambiguate the object referred to in language as "this object" or "that object".

1.3.5 Vision based gesture recognition

Computer vision supports a wide range of human tasks, including recognition, navigation, and communication. In the context of perceptual interfaces, its primary task is to detect and recognize meaningful visual information that bears messages for communication. Using computer vision to sense and perceive the user is often referred to as *Vision Based Interaction*, or *Vision Based Interfaces* (VBI). VBI demands more fine-grained analysis, where subtle facial expressions or hand gestures may be very important. Figure 1.11 shows the elements of VBI, where hand gesture recognition is one of the important elements.

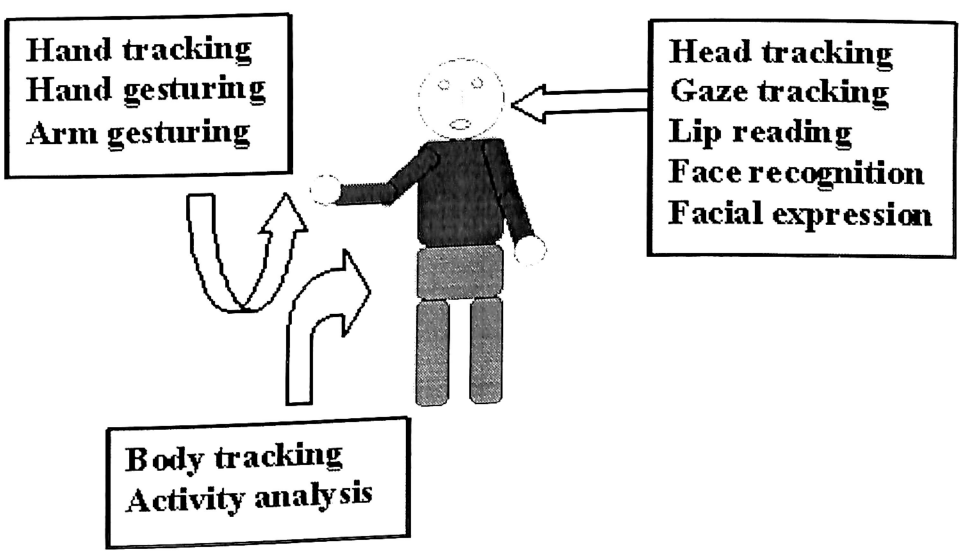


Figure 1.11: Elements of vision based interfaces (VBI).

A vision-based approach is more natural and convenient than a glove-based approach. It is easy to be deployed and can be used anywhere in the field of view of a visual camera.

TH-1870_02610203

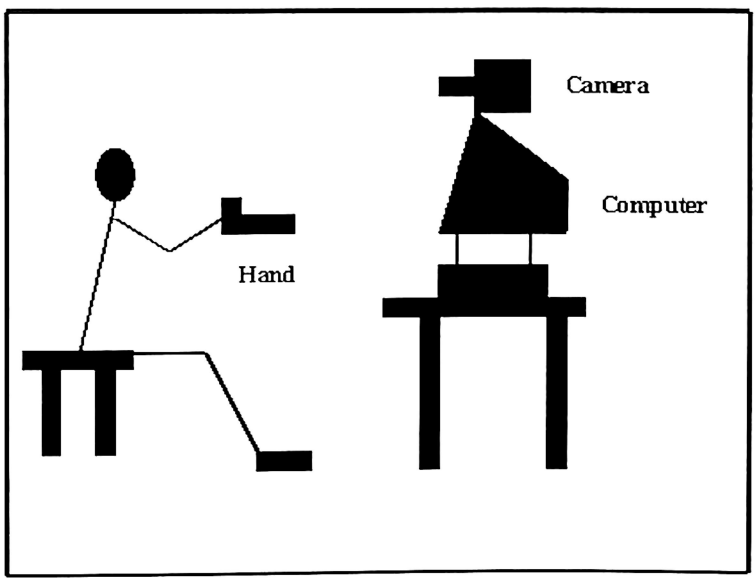


Figure 1.12: Vision based hand gesture recognition.

The operator does not need to master special hardware. Vision also allows a variety of gestures to be used, since it is basically software based. Yet it is more difficult due to the limitations of today's computer vision in handling a highly non-convex and flexible object such as a human hand. Several different approaches for vision based HCI have been proposed so far [Rehg & Kanade (1994), Maggioni (1995), Freeman & Weissman (1995), Nam & Wohn (1997), Landay & Myers (1998) and Segen & Kumar (1998)]. As shown in Figure 1.12, the straightforward approach to vision-based gesture recognition is simply to acquire visual information about a person under a certain environment and try to extract the necessary gestures. This approach has to perform the following steps: segmentation of the moving hand from a very complex environment, analysis of hand motion, tracking of hand position relative to the environment, recognition of hand postures, etc. Each of these tasks is a different problem in itself. To lower the burden, some systems use passive markers or marked gloves. The others use restrictive setups like uniform background, very limited gesture vocabulary, or just a simple static posture analysis. Some of these methods developed are given below.

Markers are usually placed on the fingertips. They are coloured in such a manner that they can be easily detected using the image histogram analysis. Once the markers are de-

TH-1870_02610203

tected and tracked, the gestures can be recognized using appropriate classification schemes. Maggioni described a hand tracking system (called Gesture-Computer) that is based on the use of a specially marked glove [Maggioni (1995)]. The glove has two slightly off-centered, differently coloured circular regions. Using single camera images, Gesture-Computer computes several image geometry parameters based on the first and second moments and uses them to estimate hand positions and orientations. Rehg & Kanade (1994) proposed a complete hand gestural interface, called DigitEyes, applicable in a relatively uncluttered background. With DigitEyes, finger tip and link parameters can be extracted from either 2D or 3D video images using edge-based techniques. These parameters are then applied to a 3D cylindrical kinematic model of the human hand with 27 degrees of freedom. Hand/arm silhouettes are one of the simplest yet widely used features. Silhouettes may be easily extracted from local hand/arm images in restricted background setups. In the case of complex backgrounds, techniques such as the colour histogram analysis is employed. In his VideoPlace, VideoDesk and VideoTouch, Krueger used silhouettes to analyze images and identify users body parts [Krueger (1991), Krueger (1993)]. Davis and Shah used simple markings on the fingertips for tracking the finger movements [Davis & Shah (1994a)]. Starner and Pentland designed colour gloves to track both hands to recognize signs from the American Sign Language (ASL) [Starner & Pentland. (1995)]. Segen and Kumar used some edge-based techniques to extract local features such as "peaks" and "valleys" from hand posture images [Segen & Kumar (1998)]. Gestures are then classified based on these local features.

Some systems have taken a speech-vision combined approach [Thorisson (1994), Maes *et al.* (1997), Vogler & Mextaxas (1997) and Vogler & Metaxas (1998)]. Such a multimodal approach is promising in offering a more natural human-computer interface.

1.3.6 Advantages of vision based gesture recognition

Using computer vision in human-computer interaction can enable interaction that is difficult or impossible to achieve with other modalities. Visual information is clearly important in human-human communication, as meaningful information is conveyed through identity, facial expression, posture, gestures, and other visually observable attributes. Therefore, intuitively it is possible to have natural human-computer interaction by sensing and perceiving

TH-1870_02610203

these visual cues from video cameras appropriately placed in the environment.

The main advantage with vision-based gesture recognition is that computer vision generally requires cheap input devices. A digital camera can be integrated into a single chip. Mass-production is therefore much easier to realize than for other input-devices with mechanical parts, such as the cyber-glove. Also, the cost for image processing hardware can be saved, as the main processors of most computers are now fast enough to take over this task themselves. More importantly, computer vision is versatile. While other input devices such as mouse, joysticks and track-pads are limited in scope to a specific function, computer vision offers a whole range of possible future applications not only in the field of human-computer interaction, but also in areas such as user authentification, video conferencing and distant-learning. Another important advantage of computer vision is its non-intrusiveness. Cameras are open input devices which do not need direct contact with the user to sense actions. The user can interact with the computer without wires and without manipulating intermediary devices.

1.4 Difficulties in Hand Gesture Recognition

The ability for computers to visually recognize hand gestures is essential for future HCI. However, vision-based recognition of hand gestures, especially dynamic hand gestures, is an extremely challenging interdisciplinary project due to the following three reasons:

1. Hand gestures are rich in diversities, multiple-meanings, and have space-time varieties;
2. Human hand is a complex non-rigid object; and
3. Computer vision itself is an ill-posed problem

The difficulties associated with hand gesture recognition are explained below.

1.4.1 Difficulties due to nature of the hand

Estimation of hand configuration is extremely difficult due to the high degrees of freedom and the problem of occlusion. Even obtrusive data gloves are not able to acquire the hand

TH-1870_02610203

state perfectly. Compared to sensors for glove-based recognition, computer vision (CV) methods are generally at a disadvantage. For monocular vision, it is impossible to know the full state of the hand unambiguously for all hand configurations, as several joints and finger parts may be hidden from the camera's view. Applications in VBI need to keep these limitations in mind and focus on gestures which may not require full hand pose information. Generic hand detection in unconstrained settings is a largely unsolved problem. In view of this, systems often use colour segmentation, motion flow, background subtraction techniques, or a combination of these to locate and track hands in images.

1.4.2 Difficulties due to gesture parameter variation

The problem of variation in gesture may be attributed to the signing person himself. There exists spatio-temporal variation when a user performs the same gesture at different times. Every time the user performs a gesture the shape and the speed of the gesture generally varies. Even if the same person tries to perform the same sign twice, small variation in speed and position of the hands will definitely occur. Moreover, as shown in Figure 1.13, the shape of the hand is another variable parameter. The system must accommodate these variations while still maintaining an accurate representation of the gestures.

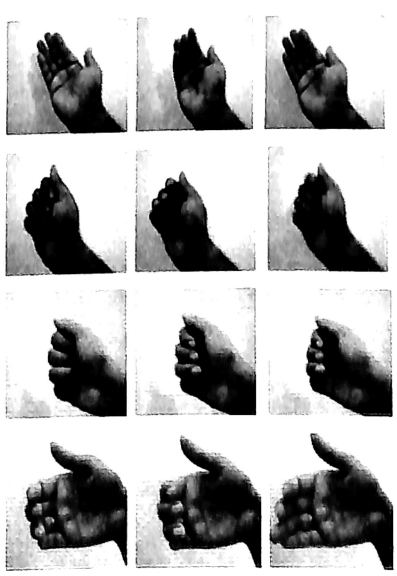


Figure 1.13: Variation of hand shapes.

TH-1870_02610203

1.4.3 Difficulties in continuous gesture recognition

The task of locating meaningful patterns from a stream of input signal is called *pattern spotting* [Rose (1992)]. *Gesture spotting* is one of the challenging aspects in the field of gesture recognition, where it is required to detect the start point and the end point of a gesture pattern in a continuous stream of video. This difficulty is due to two aspects of signal characteristics: segmentation ambiguity [Takahashi *et al.* (1992)] and spatio-temporal variability [Baudel & Beaudouin-Lafon (1993)]. The segmentation ambiguity may be attributed to the non-gestural hand movements between different gestures resulting in movement epenthesis. This creates problem in determining the beginning and end of a gesture during continuous gesturing. As the user's hand goes from the end of one gesture to the start of another gesture it makes an intermediate movement linking these two gestures. For successful recognition, the system must be able to detect sign boundaries automatically while discarding movement epenthesis. Moreover, a gesture sign is affected by the preceding and the subsequent sign. This is called co-articulation. The other difficulty in gesture spotting comes from the fact that the same gesture varies in shape and duration from one signer to another. The variation also exists even when the same gesture is made by a particular signer at different times. An ideal gesture recognizer should be capable of extracting gesture segments correctly from a continuous input video sequence and match them with reference patterns or templates allowing a wide range of spatio-temporal variability.

1.4.4 Difficulties associated with image processing techniques

Some major difficulties associated with image processing techniques during gesture recognition are as follows:

- A gesture recognition system generally uses a single colour video camera in order to minimize the necessary hardware components and calibrations. Using a single colour video camera and consequently the projection of the 3D scene on a 2D plane results in the loss of depth information. Therefore, reconstruction of the 3D trajectory of the hand in space is not possible.
- Illumination change in the testing environment is a major problem during hand image

TH-1870_02610203

segmentation.

- For reliable recognition, hand image segmentation via background motion compensation and accurate motion estimation of the hand is an essential requirement. Unfortunately, like other image processing problems, hand tracking and segmentation in a cluttered background is a critical problem in hand gesture recognition. In most occasions, the background is not a simple background. Also, the background may contain other body parts captured by the camera. Moreover, the occlusion due to other body parts or self occlusion may cause problems during segmentation.
- The processing of a large amount of image data is time consuming and so real- time recognition may be difficult.

1.5 Applications of Hand Gestures

Generally one can speak of two groups of gesture based applications: gesturing with the purpose of providing multidimensional control and gesturing as a symbolic language. Both groups aim at rendering the computer transparent to the user. A short description of different hand gesture based applications is given below.

1.5.1 Application in human computer interaction

As discussed earlier, gesture recognition plays an important role in man-machine interaction. The use of hand gestures has become an important part of HCI in recent years [Aggarwal & Cai (1997), Kjeldesen & Kender (1996), Pavlovic *et al.* (1997), Malkawi & Srinivasan (2005) and Gibet *et al.* (2005)]. Compared to many existing interfaces, hand gestures have the advantages of being easy to use, natural and intuitive. For HCI to be truly natural, technology to understand both speech and gesture together must be developed. Despite the importance of complex hand gestures in normal human-to-human interaction, most research to date in HCI and most virtual environment technology focus on the development of gesture based systems where gestures tend to be less ambiguous, less spontaneous and natural, more learned, and more culture-specific.

TH-1870_02610203

Human-robot interaction has recently drawn increased attention. Autonomous mobile robots can recognize and track a user, understand his verbal commands, and take actions to serve him [Huang & Weng (2004)] .

1.5.2 Virtual reality

Virtual reality (VR) is a technology which allows a user to interact with a computer-simulated environment, be it a real or imagined one. The human hand serves as an ideal manipulation device for virtual reality. By mapping the hand's position in the real world onto the position of a virtual hand in a virtual world it is possible to create the illusion that a person can manipulate virtual objects. This allows a person to be situated in a virtual world and interact with it through their gestures. The glove environment has been used in computer games with the purpose of giving game programmers new input alternatives. Zeller *et al.* presented a visual environment for very large scale biomolecular modeling application [Zeller. *et al.* (1997)]. This system permits interactive modeling of biopolymers by linking a 3D molecular graphics and molecular dynamics simulation program. Hand gestures serve as the input and controlling device of the virtual environment. In [Berry (1998)], Pavlovic and Berry integrated controlling gestures into the virtual environment BattleField, in which hand gestures are used not only for navigating the virtual environment, but also as an interactive device to select and move the virtual objects in the BattleField .

1.5.3 Telepresence

In any remote or hostile environment, manually controlled systems may be required to perform complex manipulation. Often it is not possible for the human operator to be physically close to the machine. Telepresence deals with the ways of providing a remote operator with as much support as possible. The support may be mapping the movements and pose of the operators arm to a robot arm. The real-time ROBOGEST system constructed at the University of California, San Diego, presented a natural way of controlling an outdoor autonomous vehicle by use of a language made of hand gestures [Katkere *et al.* (1994)]. Areas where telepresence shows prospects include space and undersea missions, in maintenance

TH-1870_02610203

of nuclear reactors and medicine [Hirche & Buss (2004)]. Major requirement in most telepresence systems is the ability to work in realtime and the need for feedback to the operator.

1.5.4 Automatic sign language recognition

Sign languages are natural languages that use different means of expression for communication in everyday life [Pedersen (1991)]. More particularly, it is the only means of communication for the hearing impaired. Thus, it offers enhancement of communication capabilities among normal beings and provides replacement for speech among deaf and mute people. Because of this, automatic sign language recognition has attracted vision researchers for long. Examples of some sign languages are the American Sign Language, the British Sign Language, the native Indian Sign Language, the Japanese Sign Language, and so on. Generally, the semantic meanings of the language components in all these sign languages differ, but there are signs with a universal syntax. For example, a simple gesture with one hand expressing 'hi' or 'goodbye' has the same meaning all over the world and in all forms of sign languages. In a sign language, the signs are generated by combinations of hand motions and finger gestures, frequently augmented with mouth movements according to the spoken language. Hand motions are distinguished from one sign to another by the spatial motion pattern, the speed, and in particular by the body parts that the signer touches at the beginning, during or at the end of a sign. In addition to the hand movement, the finger configuration during the slower parts of the hand movements also provides significant meaning to a gesture.

Recognition of a sign language is very important not only from the engineering point of view but also for its impact on human society. A functioning sign language recognition system can provide an opportunity for the deaf to communicate with hearing people without the need for an interpreter. It can be used to generate speech or text making the deaf more independent. Unfortunately, there has not been any system with these capabilities so far. Research has been limited to small-scale systems capable of recognizing a small subset of a full sign language. The reason for this is the difficulty in recognizing a full sign language vocabulary – recognition of gestures representing words and sentences undoubtedly is the most difficult problem in the context of gesture recognition research.

TH-1870_02610203

1.6 Hand Gesture Recognition: The State of the Art

As explained earlier, gesture recognition is the task in which the data analyzed from the visual images of a gesture is recognized as a specific gesture. The first phase in a recognition task is choosing a model of the gesture. As discussed in Section 1.2.2, the mathematical model should consider both the spatial and temporal characteristics of the hand and hand gestures for accurate recognition. The approach used for modelling plays an important role in the nature and performance of gesture interpretation. Once the model is decided, image features are extracted from one or more input video streams and the model parameters are computed in an analysis stage. These parameters constitute some description of the hand pose or trajectory and depend on the modelling approach used. Some of the important problems involved in this analysis stage are hand localization, hand tracking, and selection of suitable image features. The computation of model parameters is followed by gesture recognition in which the parameters are classified and interpreted in the light of the accepted model and the rules imposed by some grammar derived through training. Vision-based hand gesture recognition generally involves the following steps.

- Image acquisition: The process of getting the gesture images/videos into the system.
- Preprocessing: The process of enhancing the relevant features in the input image/video. This step involves one or more of the following processes –
  - Tracking of the gesturing hand;
  - Segmentation of the hand from the stationary background.
  - Normalization of the hand image (Hand must be normalized so that it is aligned in the same way every time independent of rotation, translation and scaling).
- Representation: The process of representation of the data in a suitable manner.
- Classification: The process of classifying the gestures via matching of the shape, pose and position of the hand.
- Post-processing: The process of evaluating the result of the classification.

In the following, we discuss different approaches developed in various literatures for gesture analysis (segmentation and feature selection) and gesture recognition.

TH-1870_02610203

1.6.1 Capturing the hand motion

Capturing the hand motion involves finding the global and local motions of the moving hand. Highly articulate human hand motion consists of the global hand motion and the local finger motion, which may be expressed as $\mathbf{M} = [\mathbf{M_G}, \mathbf{M_L}]$, where $\mathbf{M}$ is the hand motion, $\mathbf{M_G}$ is the global motion and $\mathbf{M_L}$ is the local motion [Wu & Huang (2001)]. Global hand motion that presents large rotation and translation may be written as $\mathbf{M} = [\mathbf{R}, \mathbf{t}]$, where $\mathbf{R}$ and $\mathbf{t}$ are rotation and translation, respectively. Local hand motion is articulate, and *self-occlusion* makes its detection and tracking difficult. Local hand motion may be parameterized with the set of joint angles (or hand state) as $\mathbf{M_L} = [\Theta]$ where Θ is the joint angle set. Consequently, hand motion may be expressed as $\mathbf{M} = [\mathbf{R}, \mathbf{t}, \Theta]$. One important issue in this context is how to track reliably the global hand motion from a given gesture video sequences. As discussed below, several techniques can be used for hand tracking in a long gesture video sequence and subsequently some useful features may be extracted for recognition.

Hand localization and feature detection

Hand localization means locating hand regions in image sequences, *i.e.*, the hand is to be segmented out from the background. As discussed earlier, meaningful gestures are represented by temporal hand movements and/or hand postures. Hand postures express certain concepts through hand configurations, while temporal hand gestures represent certain actions by hand movements. Sometimes, hand postures act as special transition states in temporal gestures and supply a cue to segment and recognize temporal hand gestures.

The feature selection stage is concerned with the detection of features which are used for the estimation of parameters of the chosen gestural model. Hand image features can be geometric features such as points, lines, contours, and silhouettes [Kuch & Huang (1995)]. Fingertip is one of the frequently used features, because the positions of fingertips are almost sufficient to recognize some gestures due to the highly constrained hand motion [Lee & Kunii (1995)]. Colour markers are often used to track the 3D positions of fingertips [Davis & Shah (1994b)]. Some researchers have also estimated the positions and orientations of fingertips by fitting a 3D cylinder to the images. Line fitting is another frequently

TH-1870_02610203

used technique to detect the fingertips [Rehg & Kanade (1995)].

Hand tracking

For hand gesture recognition, it is necessary that the static and/or dynamic configuration of the human hand is measured by the machine, thereby extracting gesture features to be used in classification. As discussed in Section 1.3.3, initial attempts to measure hand configuration resulted in mechanical devices, *e.g.*, glove-based devices, that directly measure hand/arm joint angles and their spatial positions. For example, Fels and Hinton used data gloves and Polhemus sensors to extract 3D hand location, velocity, and orientation. Feature vectors were then formed to represent hand gestures and were used to train a multilayer neural network for translating hand gestures to synthesized speech [Fels & Hinton (1993, 1997)]. Some other Polhemus/Glove-based hand gesture recognition techniques include [Quam (1990), Sturman & Zeltzer (1994), Kong & Ranganath (2004) and Kevin *et al.* (2004)].

Awkwardness in using gloves is overcome by using vision-based non-contact interaction techniques. These use colour-based vision segmentation, silhouettes or edges to track the hands and fingers. Most of the works on vision-based gestural HCI have mainly been focused on the recognition of static hand gestures or postures. But, hand gestures are in general dynamic actions where the motions of the hands convey as much information as their postures do. So, appropriate interpretation of dynamic gestures on the basis of hand movement in addition to shape and position is necessary.

All vision based approaches to hand/finger tracking require to locate hand regions in video sequences. Skin colour offers an effective and efficient way to segment out hand regions. Colour cues are applicable because of the characteristic colour footprint of the human skin. Most of the colour segmentation techniques rely on histogram matching [Ahmad (1994)] or employ a simple look-up table approach based on the training data for the skin and possibly its surrounding areas [Quek *et al.* (1995), Kjeldsen & Kender (1996) and McKenna & Morrison (2004)]. However, many of these techniques are plagued by some difficulties such as large variation in skin tone, unknown lighting conditions and dynamic scenes. This frequently results in undetected skin regions or falsely detected as

TH-1870_02610203

nonskin regions. An approach based on the 2D locations of fingertips and palms was used by Davis and Shah as early as in 1994 [Davis & Shah (1994a)]. Similarly, some of the gesture recognition applications use uniquely colored gloves or markers on hands/fingers [Cipolla *et al.* (1993), Lee & Kunii (1995) and Maggioni (1993)]. The use of background restriction or coloured gloves makes it possible to localize the hand efficiently and even in real-time, but imposes the obvious restriction on the user and the interface setup.

The motion cue is also commonly applied for segmentation and is used in conjunction with certain assumptions about the gesturer. In the HCI context, these assumptions include for example the following:

- Only one person makes gestures at any given time
- The signer is usually stationary with respect to the background. Hence, the main component of motion in the visual image is usually the motion of the arm/hand of the gesturer and can thus be used to segment out the moving parts of the gesturer.

This localization approach is followed in [Freeman & Weissman (1995), Quek (1995)]. The disadvantage of the motion cue approach lies in its assumptions. While the assumptions hold over a wide spectrum of cases, there are occasions when more than one gesturer is active at a time or the background is not stationary.

To capture articulate hand motion, both global hand motion and local finger motion should be determined from video sequences. It is a challenging problem to capture and analyze the hand motion, because the hand is highly articulate. Simple 2D motion trajectories can be extracted from the image sequences using 2D tracking that also supplies localized information such as the hand bounding boxes and the centroid of hand blobs. In some cases, these 2D features are sufficient for gesture recognition. There are many 2D tracking algorithms such as those based on colour tracking, motion tracking, template matching, blob tracking, and multiple cues integrating. Although 2D tracking gives the positional information of the hand, some recognition applications still need more features such as the hand orientation and the hand shape. This is taken care of in 3D tracking approaches that try to locate the hand in 3D space by finding the 3D position and orientation of hand. However, since the hand cannot be treated as a rigid object, it is very difficult to estimate the hand

TH-1870_02610203

orientation. The 3D position of the hand may be located by stereo camera or model-based approaches. In the 3D model-based approach, the hand configuration is estimated by taking advantage of 3D hand models [Kuch & Huang (1995), Lee & Kunii (1995), Rehg & Kanade (1995), Heap & Hogg (1996), Pavlovic *et al.* (1997) and Wu & Huang (1999a)]. With a single calibrated camera, local hand motion parameters can be estimated by fitting the 3D model to the observation images. 3D model based methods take advantage of *a priori* knowledge built into the 3D models. This approach aligns a 3D model to images or even range data by estimating the parameters of the model. In 3D model-based methods, image features may be looked as the image evidence or image observation of a 3D model that is projected onto the 2D image plane. A 3D model with different parameters will produce different image evidence. Model-based methods recover the joint angles by minimizing the discrepancy between the image feature observations and the projection of the 3D model. However, they lack simplicity and computational efficiency. An alternative approach is appearance-based modeling [Wu & Huang (2000a)]. It emphasizes the analysis of hand shapes in images, in which 2D deformable hand shape templates are used to track a moving hand in 2D. However, this method is insufficient to recover full articulations, because it is difficult to infer finger joint angles based on appearances alone. Although it is easier for the appearance-based approach to achieve user-independence than the 3D model-based approach, there are two major difficulties associated with this approach, *viz.*, automatic feature selection and training data collections [Wu & Huang (2001)].

Another important approach for hand tracking is the prediction algorithm combined with *Kalman filtering* and application of probabilistic reasoning for final inference [Zieren *et al.* (2002), Wu & Huang (2001) and Ramamoorthy *et al.* (2003)]. Kalman filtering is a state estimation method based on Gaussian distribution. Unfortunately, it is restricted to situations where the probability distribution of the state parameters is unimodal. It is inadequate in dealing with simultaneous multi-modal distributions in the presence of occlusion, cluttered background resembling the tracked objects, etc. Moreover, this approach requires a fine initial model and also demands additional computations whenever there is any rotation and change in shape of the model. Kalman filtering is based on small motion assumption that often fails to hold in hand motion [Wu & Huang (2001)]. The *Conden-*

TH-1870_02610203

sation algorithm [Isard & Blake (1996)] has shown to be a powerful alternative. It is a conditional density propagation method for visual tracking. Based upon sampling the *a posteriori* distribution estimated from the previous frame, these samples iteratively propagate to successive images. By combining a tractable dynamic model with observations, it can accomplish highly robust tracking of object motion. It usually requires a relatively large number of samples to ensure a fair maximum likelihood estimation of the current state.

1.6.2 Static hand posture recognition

Since hand postures not only can express some concepts, but can also act as special transition states in temporal gestures, recognizing or estimating human hand postures is one of the main topics in gestures recognition.

One approach is the 3D model-based approach, in which the hand configuration is estimated by taking advantage of 3D hand models [Kuch & Huang (1995), Lee & Kunii (1995), Rehg & Kanade (1995), Heap & Hogg (1996), Wu & Huang (1999a)]. Because hand configurations are independent of view directions, these methods can directly achieve view-independent recognition. Different models use different image features to construct feature-model correspondences. Joint angles can be estimated by minimizing a projected surface model and some image evidences such as silhouettes in the light of analysis-by-synthesis [Davis & Shah (1994b), Kuch & Huang (1995), Lee & Kunii (1995)]. This approach needs good surface models and the process of projection and comparison is expensive. Alternatively, point and line features are employed in kinematical hand models to recover joint angles [Rehg & Kanade (1995), Wu & Huang (1999a)]. Hand postures can be estimated accurately if the correspondences between the 3D model and the observed image features are well established. Physical models and statistical models were also employed to estimate hand configurations [Heap & Hogg (1996)] . Many current methods require reliable feature detection, which is plagued by self-occlusion. Another drawback is that it is not easy to achieve user independence, because 3D models should be calibrated for each user.

Triesch and Malsburg employed the elastic graph matching technique to classify hand postures against complex backgrounds [Triesch & Malsburg (1996)]. Hand postures are

TH-1870_02610203

represented by labeled graphs with an underlying two-dimensional topology. Attached to the nodes are jets, which are a sort of local image description based on *Gabor filters*. Gabor filters are defined by harmonic functions modulated by a Gaussian distribution [Kamarainen *et al.* (2006)]. This approach can achieve scale-invariant and user-independent recognition, and it does not need hand segmentation. Since using one graph for one hand posture is insufficient, this approach is not view-independent.

Nolker and Ritter detected the 2D location of fingertips by the Local Linear Mapping neural network (LLN)[Nolker & Ritter (1998)]. The 2D locations are mapped to 3D position by the Parametric Self-Organizing Map (PSOM) neural network. The PSOM has the ability to perform an associative completion of fragmentary inputs. By this means, their approach can recognize the hand pose under different views.

Because the estimation of hand joint angles is difficult, an alternate approach is the appearance-based approach which aims at characterizing the mapping from the image feature space to the possible hand configuration space directly from a set of training data [Rosales & Sclaroff (2000), Wu & Huang (2000b)]. This approach often involves learning techniques.

1.6.3 Dynamic gesture recognition

There are some similarities between dynamic gestures and speech so that some techniques in speech may be applied to gestures. However, dynamic gestures are more complicated than speech. Some low-level movements can be recognized using dynamic models. Again, some gesture semantics can be exploited to recognize high-level activities [Wu & Huang (1999b)]. Example-based learning methods can also be used for this purpose.

Gesture recognition by modeling the dynamics

Modelling the low-level dynamics of human motion is important not only for human tracking, but also for human motion recognition. It serves as a quantitative representation of simple movements so that they can be recognized in a reduced space by the trajectories of motion parameters. Models of low-level dynamics are generally not sufficient to represent more complicated human motions. Some low-level motions can be represented by simple

TH-1870_02610203

dynamic processes, in which the Kalman filter is often employed to estimate, interpolate and predict the motion parameters. This simple dynamic model is not sufficient to model most cases of human motion, and the Gaussian assumption of the Kalman filtering is not satisfied in practical cases. A conditional density propagation (Condensation) algorithm proposed by Isard and Blake adopts the stochastic differential equation to describe complex motion models, and combines this approach with deformable templates to cope with people tracking [Isard & Blake (1996)]. Later, this idea was extended by Black and Jepson to recognize gestures [Black & Jepson (1998a) and Black & Jepson (1998b)].

Pentland and Liu represented human behavior by a complex, *multistate model* [Pentland & Liu (1999)]. They used several alternative models to represent the human dynamics, one for each class of response. *Model switching* is based on the observation of the state of the dynamics. This approach produces a generalized maximum likelihood estimate of the current and future values of the state variables. Recognition is achieved by determining which model best fits the observation.

Rittscher and Blake proposed a technique in which the idea of model switching and condensation are combined [Rittscher & Blake (1999)]. They used mixed discrete/continuous states to couple perception with classification. In this method, the continuous variable describes the motion parameters and the discrete variable labels the class of the motion. Finally, an ARMA model is used to represent the dynamics. This approach can achieve automatic temporal sequence segmentation. There is also some work dealing with specific gestures. Cohen *et al.* used *dynamic models* to represent circle and line gestures to generate and recognize basic oscillatory gestures such as the crane control gestures [Cohen *et al.* (1996)].

Gesture recognition by modeling the semantics (state-based approaches)

Modelling the dynamics alone may not be sufficient for dynamic hand gesture recognition. The *Finite State Machine* (FSM) is a usually employed technique to handle this situation. The most important feature of an FSM is its state-transition function. The states are used to decide which reference sequence matches with the test sequence. It allows a more sophisticated analysis of data with spatio-temporal variability. Some of the state-based approaches

TH-1870_02610203

were proposed in [Davis & Shah (1994b), Bobick & Wilson (1997) and Hong *et al.* (2000)].

Davis and Shah introduced a simple real-time method for dynamic gesture recognition by tracking the trajectory of each fingertip, wearing a marked glove, and using the extracted motion as basis for recognition [Davis & Shah (1993) and Davis & Shah (1994a)]. The displacement and direction of each fingertip are computed, and by averaging the information obtained for several samples of the same gesture a model database is created. The database of motion codes is then compared to the motion code of a new gesture, and recognized if present in the database. An FSM is used to model all the gestures. Davis and Shah characterized a gesture by four phases: (1) static start position, for at least 3 frames; (2) smooth motion of the hand and fingers until the end of a gesture; (3) static end position for at least 3 frames; and (4) smooth motion of the hand and fingers back to the start position. A major disadvantage of the system is that the user is constrained to follow these four phases. If the start and stop positions are not correctly detected, the FSM will work asynchronously and fail to recognize the gestures.

Similar to Davis and Shah, Bobick and Wilson made the assumption that the variability of a gesture is captured implicitly by a number of examples of that particular gesture [Wilson & Bobick (1995a) and Wilson & Bobick (1995b)]. They represented a gesture by summarizing a set of local motion trajectory variances of a number of training examples of each gesture. The actual recognition is performed by determining whether a new trajectory is consistent with those obtained from training.

An important state-based approach to dynamic gesture recognition is by using the *hidden Markov models* (HMM) which has been proved to be the most efficient tool, as of now, in classifying dynamic data. The HMM is a doubly stochastic state machine, a probabilistic network with hidden and observable states [Brand *et al.* (1997)]. The hidden states "drive" the model dynamics – at each time instance, the model is in one of its hidden states. Transitions between the hidden states are governed by probabilistic rules [Rabiner & Juang (1993)]. The observable states produce outcomes during hidden state transitions or while the model is in one of its hidden states. Such outcomes are measurable by an outside observer. The outcomes are also governed by a set of probabilistic rules. The HMM involves two stages: training and classification. In the training stage, the number of states of an

TH-1870_02610203

HMM must be specified, and the corresponding state transition and output probabilities are optimized in order that the generated symbols conform to the observed image features of the examples within a specific movement class. In the matching stage, the probability with which a particular HMM generates the test symbol sequence corresponding to the observed image features is computed.

An HMM-based hand gesture recognition system was developed by Yamato *et al.* who made use of the mesh features of the 2D moving human blobs such as motion, colour and texture, to identify the human behavior [Yamato *et al.* (1992)]. In the learning stage, the HMMs are trained to generate symbolic patterns for each action class, and the optimization of the model parameters is achieved by forward-backward algorithm. Given an image sequence, the output result of forward calculation is used to guide action identification during recognition. However, the selected mesh features do not match the human posture space and are sensitive to positional displacement.

Schlenzig *et al.* described and implemented a telepresence system, called ROBOGEST, using a *recursive filter* applied on a predefined set of gestures to control a moving vehicle [Katkere *et al.* (1994), Schlenzig *et al.* (1995)]. The predefined gestures to control the vehicle included those to stop, toggle direction, accelerate, and reset. The gestures were modeled using an HMM with the state representing the gesture sequences with the observations being the current static hand poses. At each time step, the recursive filter updates its estimate of what gesture is currently being performed based on the extracted pose information.

Lee and Xu also developed an interactive gesture recognition system based on the HMM [Lee & Xu (1996)]. Their focus was on designing a system that is able to learn new gestures and accordingly update the currently available gesture model in the online mode. Such a system can be applied for a broader range of applications. The system uses DataGlove for detecting hand movements. In another work, Pentland and Liu used the HMM to model the state transitions among a set of dynamic models [Pentland & Liu (1999)]. Bregler also took the same approach in [Bregler (1997)]. The HMM has the capacity for not only modelling the low-level dynamics, but also the semantics in some gestures. In [Stoll & Ohya (1995)], Stoll and Ohya employed the HMM to model semantically meaningful human movements.

TH-1870_02610203

In this, one HMM is learned for each motion class. The data used for modelling the human motions is an approximate pose derived from an image sequence. Nam and Wohn presented an HMM based method to recognize some controlling gestures [Nam & Wohn (1996)]. Their approach takes into account not only hand movement, but also hand postures and palm orientations.

Darrell and Pentland introduced a *hidden state reinforcement learning paradigm* based on the partially observable Markov decision process to gesture recognition by which an active camera is guided [Darrell & Pentland (1996)]. Wilson and Bobick extended the standard HMM method to include a global parametric variation in the output probabilities of the HMM to handle parameterized movements. Using a linear model to derive the theory, an *expectation-maximization* (EM) method for training the parametric HMM was formulated. During testing, the parametric HMM simultaneously recognizes the gesture and estimates the quantifying parameters [Wilson & Bobick (1998)]. They presented results on two different movements – a size gesture and a point gesture, and showed robustness with respect to noise in the input features. Later, several other HMM-based hand gesture recognizers were developed in [Vogler & Metaxas (1998)], Chen *et al.* (2003) and Ramamoorthy *et al.* (2003)]. But, since the output probability of feature vectors of each state in an HMM is unique, the HMM can handle only piecewise stationary processes which are not adequate in proper gesture modelling. When the Markov condition is violated, conventional HMMs fails. Thus, HMMs are ill-suited to systems that have compositional states [Brand *et al.* (1997)].

Other techniques

As discussed earlier, most of the model based methods use DataGlove to read the hand configuration. DataGlove provides data represented in a way that makes it easy to model. In model-based recognition, the recognition process typically fits the input data to a model and therefore belongs to the group of structure based methods. Obvious choices for a model are the joint angle between the palm and each finger, the shape of the palm, the location of the hand relative to the user, etc. Rehg and Kanade described a real-time model-based dynamic gesture recognition [Rehg & Kanade (1993)]. Using simple features extracted

TH-1870_02610203

from natural images of a human hand they were able to extract the pose of a kinematic 27 degrees of freedom articulated hand. The solution consisted of three basic components: a state model, feature measurements, and state estimation. The state model is composed of the pose of the palm (seven states) and the joint angles of the fingers (five states for the thumb four states for the rest) giving a 28-dimensional state vector. The features for the hand consist of finger links and fingertips. A state estimate (prediction) is calculated for each image to obtain a state that best fits the measured features. Features are matched by a nearest-neighbor classifier. Since the initial pose of the hand must be known for tracking the hand is initially placed in a known position and pose which initialize the system.

Takahashi and Kishino created a user-dependent glove environment system that was able to recognize 34 gestures out of 46 Japanese Kana alphabet using a joint angle and hand orientation coding technique [Takahashi & Kishino (1991)]. The user makes each of the 46 gestures 10 times to provide data for the *Principal Component Analysis*(PCA) and cluster analysis. The PCA can be seen as a mix of two methods: template matching and statistical classification. The PCA proved suitable for real-time application and is a way of representing data before classification. Most recently, Kato *et al.* proposed an efficient representation of hand motions by the *Independent Component Analysis* (ICA) [Kato *et al.* (2006)]. The ICA basis vectors represent local features, each of which corresponds to the motion of a particular finger. In the ICA-based model, hand poses can be represented by five parameters with each parameter corresponding to a particular finger motion. This representation is more efficient in modeling hand motions for tracking and recognizing hand-finger gestures in an image sequence.

As discussed earlier, vision based interfaces eliminate the difficulties posed by sensory devices. The simplest method for recognizing gestures is probably *template matching*, mainly used with the vision environment [Kaneko & Hori (2003)]. It is an appearance-based method with no representation stage, since an input image is used directly in the decision process. The classification is based on a similarity function which measures the similarity between the input image and a number of stored template images of the gestures to be recognized. The similarity function is often the sum of squared distances or a normalized correlation. The input image is classified on the basis of this similarity measure.

TH-1870_02610203

To avoid the problem of classifying an image that does not represent a gesture, an input is rejected from classification if the similarity measure falls below some predefined threshold. A drawback of template matching is its sensitivity to scaling, translation and rotation. The method is also computationally demanding and hence there is an upper limit of the number of template gestures to be matched. Another problem is that of generating the templates that will give user independence and illumination independence.

Darrel and Pentland presented a real-time glove-free system for recognizing dynamic gestures from a sequence of images [Darrell & Pentland (1994), Darrell *et al.* (1995)]. Their approach employs an automatic appearance-based method to build a set of view-models that each gesture model is generated from. In the off-line mode, the first view is chosen by the user as an image in the sequence. The next view is found by correlating the first image with the subsequent images. When the correlation falls below a preset level, a new view of the current image is created. New views are chosen until the gesture is completed. Several examples of each gesture is used to create a mean and variance of the correlations. A gesture is now described by the views that were chosen over time. The next step is to align the gestures in time; this is done by the *Dynamic Time Warping* (DTW) technique. In the online mode, a gesture will be correlated with all the views of a model with respect to time. A score plot of the new sequence compared to each of the stored views makes it possible to match a new sequence with a gesture that is already known. The method is computationally very demanding because the number of correlations needed for each sequence is proportional to the number of gestures to be recognized. Also, the normalized correlation that is used to match the 2D views is very sensitive to noise.

Freeman and Roth outlined a system for controlling the television through gestural inputs [Freeman & Roth (1995) and Freeman & Weissman (1995)]. Their idea was to exploit the television's capability of visual feedback. Based on graphics shown on the TV, the viewer sees how to move his hand to issue commands normally issued through a remote control. Their approach employed the *Orientational Histograms* in the recognition of gestures. The orientational histogram is a simple way of describing the information contained in an image. An orientational histogram may be based on the gradient of the image mapped according to its direction. The orientation is mapped into polar coordinates and compared

TH-1870_02610203

to the orientation template of each predefined gesture. The orientational histogram has one major advantage that it is insensitive to illumination changes, but also has a major disadvantage of being sensitive to rotation and translation.

The research in neural networks has shown great promise in pattern recognition. The reason for the popularity lies in the fact that once it is configured using a set of training examples it forms an appropriate internal representation and decision system. The representation is distributed across the network as a series of interdependent weights instead of conventional local data structure. Generalization is an important advantage of the system, *i.e.* to a certain degree the variability of the gestures between samples can be compensated by the network. However, neural nets have several serious drawbacks. A large amount of training examples are required to obtain a precise recognition. The training phase is computationally demanding and must be repeated if a new gesture is to be added. Another problem is to understand how the network actually obtains its weights through training. The net itself is considered as a black box in which only the number of input and output nodes are set by the designer. Boehm *et al.* used the neural network to recognize dynamic gestures [Boehm *et al.* (1994)]. One reported disadvantage in their method is the large processing time for each gesture.

Miners *et al.* presented a *graph-based* approach to understand the meaning of hand gestures by associating dynamic hand gestures with known concepts and relevant knowledge [Miners *et al.* (2005)]. They used only a small vocabulary as well as a small knowledge-base.

1.6.4 Sign language recognition

The most complicated part of dynamic hand gesture recognition is the sign language recognition as both local and global motions of the hand are necessary in addition to temporal information. Sign language recognition is a multidisciplinary research area involving pattern recognition, computer vision, natural language processing and psychology. It is a comprehensive problem because of the complexity involved in the visual analysis of hand gestures and the highly structured nature of the sign languages. Sign languages are well-structured languages with a phonology, morphology, syntax and grammar distinctive from

TH-1870_02610203

spoken languages. The structure of a spoken language makes use of words linearly, *i.e.*, one after the other, whereas a sign language makes use of several body movements parallelly in the spatial as well as in temporal space. The linguistic characteristics of a sign language are different than that of spoken languages due to the existence of several components affecting the context such as the use of facial expressions and head movements in addition to the hand movements.

In sign language recognition, there are two kinds of gestures to be recognized: one is the isolated gesture, and the other is the continuous gesture. While the presence of silence (significant hand pause) makes the boundaries of isolated gestures easy to spot, continuous sign recognition is much harder as there is no silence between the signs [Wu & Huang (1999b)]. Therefore, recognition of gestures representing words and sentences, as they do in sign languages, undoubtedly represents the most difficult recognition problem. Researches till date have been limited to small scale systems capable of recognizing only a minimal subset of a full sign language.

The earliest reported work on sign language recognition is available in [Starner & Pentland. (1995) and Starner *et al.* (1998)]. Starner and Pentland developed a glove-based system capable of recognizing a subset of the American Sign Language (ASL). In this method, tracking of both hands is done by extracting a feature vector consisting of the x and y positions of hands, the angle of axis of the least inertia (found by the first eigenvector of the hand blob), and eccentricity of the bounding ellipse. These give a rough description of the hand shape, orientation and trajectory. Through the use of the time context provided by an HMM it is possible to recognize the sentence-level ASL. There is no explicit modeling of the fingers. A disadvantage of the system is that it is user dependent.

As discussed earlier, one major problem associated with continuous hand gesture recognition is "co-articulation". There are several approaches to deal with the co-articulation problem. In one method, Vogler and Metaxas used the *context-dependent* HMMs. The idea of context-dependent HMMs is to train bi-sign or even tri-sign context dependent HMMs. They also proposed an unsupervised clustering scheme to obtain the necessary classes of "phonemes" for modelling the movements between signs [Vogler & Mextaxas (1997), Vogler & Metaxas (1998) and Vogler & Metaxas (1999)]. The ASL signs are bro-

TH-1870_02610203

ken into phonemes such as movements and holds. Phonemes are used instead of whole signs as the basic units. HMMs are trained to recognize the phonemes. Since the number of phonemes is limited, it is possible to use HMMs to recognize large vocabularies.

Liang and Ouhyoung [Liang & Ouhyoung (1998)] also used the HMM approach for recognition of continuous Taiwanese Sign Language with a vocabulary of 250 signs. The temporal segmentation is performed based on the discontinuity of the movements according to four gesture parameters namely posture, position, orientation and motion.

The *time-delay neural network* (TDNN) is an interesting approach for analyzing time-varying data [Hu *et al.* (2004)]. In the TDNN, delay units are added to a general static network. Some of the preceding values in a time-varying sequence are used to predict the next value. As larger data sets become available, more emphasis is given to neural networks for representing temporal information. The TDNN has been successfully applied to hand gesture recognition [Yang & Ahuja (1998)] and lip-reading [Meier *et al.* (2000)]. Yang and Ahuja used the TDNN to classify motion patterns [Yang *et al.* (2002)]. The TDNN is trained with a database of ASL signs where the input is the motion trajectories extracted via multi-scale motion segmentation.

Another important research area of sign language recognition is the *finger spelling* recognition. For this, finger motions have to be detected accurately. Some of the related works are given in [Jerde *et al.* (2003), Chandran & Sawa (2004) and Wu *et al.* (2005)].

Though researchers proposed different methods for sign language recognition, none of these methods are successful to address all the problems encountered in sign language recognition simultaneously. This has motivated the ongoing research in the field of gesture recognition and sign language interpretation.

1.7 Motivation of the Thesis

As discussed earlier, HCI demands for recognition of wide classes of natural gestures in a vision based platform with minimum hardware and simplified algorithmic requirements. Different algorithms for recognition of hand gesture have been proposed till date but none of them is totally successful in recognizing different types of hand gestures in a common

TH-1870_02610203

framework. The research in vision-based gesture tracking and recognition is still far from maturity [Wu & Huang (2001)] and has not yet advanced to a stage where it can be successfully deployed as a user interface in consumer-grade applications. The main challenges in this context are robustness, user independence and view independence. Vision-based recognition must achieve precision in two complementary domains – very fast tracking of the position of the hand in the 3D space and exact estimation of the configuration of the fingers in the hand. To combine these requirements in one system is a major challenge. Therefore, in order to develop a natural and reliable hand gesture interface, substantial research efforts in computer vision, graphics, machine learning, and psychology is necessary. Accordingly, this thesis looks into several aspects concerned with hand gesture recognition and aims at developing algorithms that take care of some of these aspects. The motivation behind this research work are given below.

1. Development of a hand tracking and segmentation method with better performance in a single-camera setup.
2. As discussed earlier, the hand gesture recognition problem can be categorized as follows.
    - Recognition of static hand postures.
    - Recognition of dynamic hand gestures having only local hand motions (only fingers and palm have motions while the arm does not move).
    - Recognition of dynamic hand gestures having only global hand motions (the position of the hand in the space changes due to arm motion while the fingers and palm do not make any meaningful movement).
    - Recognition of dynamic hand gestures having both local and global hand motions.
    - Recognition of continuous hand gestures in which the user continually makes gesture signs. Here a sequence of meaningful gestures are embedded in a continuous video stream with co-articulation and unintentional movements in between gestures.

Existing algorithms for recognition cannot recognize hand gestures having different

TH-1870_02610203

spatio-temporal behaviours in a common platform. These algorithms are only successful in recognizing only a particular type of hand gesture. Therefore, it is necessary to develop a scheme that will integrate different gesture recognition methods in a common system.

3. All the finite state machine representation algorithms [Davis & Shah (1994b), Bobick & Wilson (1997), Hong *et al.* (2000)] use each frame in the gesture sequence to build up the gesture model. This seems to be computationally inefficient. Since there exists large amount of temporal redundancy between frames in a video sequence, it is quite possible to obtain the gesture model from only a selected number of frames in the sequence.
4. Recognition of continuous hand gestures in the vision based setup is an interesting and open research challenge. This is due to the effect of co-articulation and the difficulties in gesture spotting. Till now very little work has been reported to cope with this problem. The methods proposed so far do not address the problems encountered in recognizing all types of continuous hand gesture in a vision based platform.
5. As discussed earlier, sign languages, although different in form, serve the same function as a spoken language. They are natural languages which are used by deaf-and-mute people all over the world. As a consequence, the development of a system for translating a sign language into a spoken language will be of great help for communication between a mute person and a normal being.

1.8 Objective of the Thesis

The goal of this research work is to build a system that uses natural gestures as a modality for recognition in a vision based setup. Our focus is to recognize dynamic hand gestures having different spatio-temporal characteristics for HCI applications.

The thesis aims at developing algorithms for tracking and segmentation of the hand with reduced computational load and better performance. We propose to borrow some of the concepts used in the MPEG-4 video standard for this purpose. Particularly, the thesis applies the concept of key video object planes for gesture summarization. This summarization

TH-1870_02610203

will be used for representation of gestures with local hand motion only by an FSM-based method. For representation of gestures having global hand motion only, a trajectory-based method will be used. These representations will be used for identifying gestures. The possibility of incorporation of the shape, the spatial position and motion of the hand in a single recognition system is to be investigated for recognizing wide classes of gestures. The other issue to be investigated is segmentation of a continuous stream of hand gestures having different spatio-temporal and motion behaviours. It will be done using hand pauses as well as motion information thereby eliminating co-articulation occurring in between gestures.

1.9 Organization of the Thesis

This thesis contains six chapters, including present one. **Chapter 2** discusses the proposed scheme for recognition of hand gestures that consists of local hand motions only. We propose a method for hand tracking and segmentation followed by construction of the finite state machine model based on shape and temporal information necessary for recognition. **Chapter 3**, presents the proposed approach for dynamic hand gesture recognition that involves only global hand motion. The method for extraction of hand motion trajectory and selection of some essential trajectory features for recognition are explained in this chapter. **Chapter 4** explains the proposed dynamic hand gesture recognition model for gestures having both local and global hand motions. For this, the hand shape information in the form of key video object plane and spatio-temporal information in the form of motion trajectory are used. We develop a scheme for determining the nature of the input gesture and subsequently apply the appropriate gesture recognition algorithm. Methods for continuous hand gesture recognition is proposed in **Chapter 5**. A discussion on different problems encountered in continuous hand gesture recognition is included in this chapter. We propose a method to determine the nature of hand movement at every instant of continuous gesturing thereby identifying gestural and co-articulation phases in the input video stream. Finally, we draw our conclusion in **Chapter 6** highlighting the strengths and shortcomings of our schemes and outlining possible extensions.

TH-1870_02610203

Chapter 2

Recognition of Hand Gestures having Local Motions only

One form of dynamic hand gestures is that in which the gestures are performed by bending one or more fingers and/or turning the palm in different orientations while the hand as a whole is held fixed in space. That is, this type of gestures consists of motions localized at the finger joints and/or the wrist while there is no global motion of the whole hand. Accordingly, we refer to gestures of this type as hand gestures having local motions only. In this chapter, we develop a novel scheme for recognizing this type of gestures. In our method, we first form the video object plane (VOP) from every frame in a given gesture sequence, where the gesturing hand is the video object. Thus, a series of VOPs is obtained from the given gesture video sequence. Next, the key VOPs (KVOPs) in the sequence are selected on the basis of the measure of change in hand pose – for a given key VOP, the next key VOP is the one in which the object (hand) changes its shape significantly. By doing so, an entire video clip is transformed into a small number of representative VOPs that are sufficient to define the input gesture. Subsequently, we model the gesture as a finite state machine (FSM), where each state is defined in terms of hand shape parameters in each key VOP and the condition for state transition is determined by the duration of a KVOP. During recognition, the states of an input gesture are matched with that of a prototype FSM available for each gesture in the vocabulary. In our experiment, the proposed method is used to identify different gestures and a recognition rate of more than 90% is obtained. This proves the effectiveness of the proposed KVOP based recognition of gestures having local motions only.

TH-1870_02610203

2.1 Proposed Scheme for Gesture Recognition

One form of dynamic hand gestures that is used in many HCI applications is one in which different gesture signs are generated by bending one or more fingers and/or turning the palm in different orientations while the hand as a whole is held fixed in 3D space. That is, this type of gesture consists of motions localized at the finger joints and/or the wrist while the hand as a whole does not move. Accordingly, we refer to gestures of this type as those having local motions only. One such gesture sequence is shown in Figure 2.1.

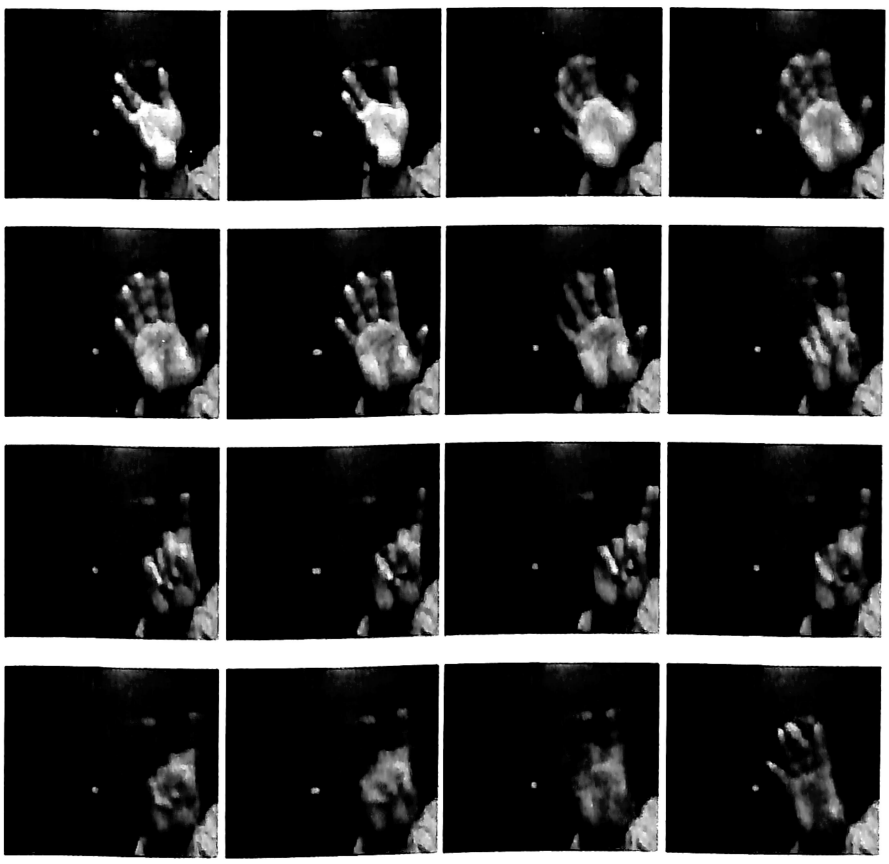


Figure 2.1: A gesture video sequence showing only local hand motions.

As we see in this figure, the position of the hand in each frame is more or less fixed and the gesture is formed only by changing the hand pose. Another common example of dynamic hand gestures having local motions only is finger spelling. A letter in a sign language is represented by a static hand pose, and a sequence of such different hand poses

TH-1870_02610203

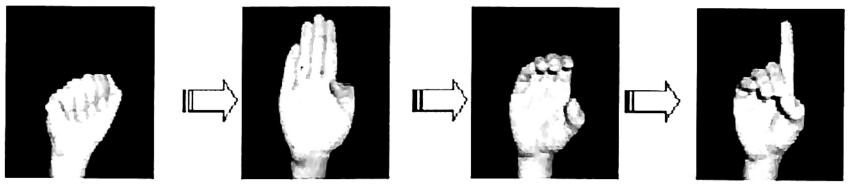


Figure 2.2: A finger spelling gesture sequence representing "ABCD" in ASL.

is used to spell a word. For example, Figure 2.2 shows making of "ABCD" in American Sign Language (ASL).

In view of the various problems posed by one or the other algorithms mentioned in the previous chapter, we propose a novel scheme for recognition of hand gestures having only local motions. The algorithm is efficient both in terms of overall system complexity and recognition accuracy. When viewed in a vision based gesture recognition system, the change in hand pose appears as deformation in hand image shape. Accordingly, we propose to identify these gestures by exploiting hand shape information. The proposed algorithm is based on the *finite state machine* (FSM) representation of gestures using key video frames in a given gesture sequence. The gesture corresponding to the input sequence is represented by a sequence of states where every state is defined in terms of hand shape parameters in each key frame. Finally, recognition is accomplished simply by matching the FSM of the input gesture to that of each gesture in the vocabulary.

In the proposed technique, we use the concept of object-based video abstraction, as used in the MPEG-4 video coding standard, for segmenting each frame in the input gesture sequence into a *video object plane* (VOP), where the gesturing hand is considered as the *video object* (VO). A binary image for the hand is subsequently derived and is tracked from one frame to the next using the *Hausdorff tracker* [Huttenlocher *et al.* (1993b), Meier & Nagan (1998)]. Next, the key VOPs in the sequence are selected on the basis of the Hausdorff distance measure, thereby transforming an entire video clip into a small set of representative VOPs that are sufficient to represent a particular gesture sequence in a summarized form[1]. The frames corresponding to the key VOPs are the key frames that best represent the con-

[1]This work has been published in *IEEE TENCON 2004* (Refer item 5 in Page 198 for details)

TH-1870_02610203

tent of the sequence in an abstracted manner. A gesture can now be defined as an ordered sequence of states, where the hand shape information in every key frame forms a state and the number of frames between one key frame and its next key frame gives the time duration for which the hand remains in that state. This constitutes a finite state machine (FSM) representing the input gesture. For recognition, the FSM representing the input gesture are matched to each of the prototype FSMs contained in the gesture database, each corresponding to one particular gesture sign in the gesture vocabulary. The matching is done on the basis of state (hand shape) and duration of each state – the input gesture is identified as the one for which the FSM matching score is maximum.

One notable advantage of the proposed scheme is its robustness to background noise inherent in Hausdorff tracking. The Hausdorff tracker can track the object very efficiently even without background filtering [Huttenlocher *et al.* (1993a), Meier & Nagan (1998)]. Furthermore, the Hausdorff tracker has the provision of tracking the hand even when it disappears momentarily from the capture range of the camera or when there is sudden change in the hand shape. The algorithm can also be modified by including background motion estimation and computation, as proposed by [Meier & Nagan (1998)], so as to extract VOPs even when the background is moving. Another advantage with Hausdorff tracking is that it is computationally more efficient than some other well known object tracking methods like Kalman filter based tracking. This is because, Hausdorff tracking requires no extra computation, such as affine transformation, on account of scaling and rotation of the video object. In this, "shape change" of the tracked object can accommodate scaling and rotation in successive frames of the gesture video sequence. The only computation required on account of shape change is the model update in each frame which is generally computationally much cheaper than affine transformation.

Compared to the HMM based systems, where the number of states and the structure of the HMM must be predefined, a gesture model is available immediately in an FSM based system. As pointed out in [Bobick & Wilson (1997)], the statistical nature of an HMM precludes a rapid training phase. While well-aligned data segments are required to train an HMM, the training data are segmented and aligned simultaneously during training to produce a finite-state machine for a gesture. Another advantage of using the FSM is

TH-1870_02610203

that it can handle patterns of different lengths and states. An FSM can adapt quickly to accommodate spatio-temporal variability during the training phase and is therefore robust to any variation in length of the input gesture sequences.

The major contribution here is that the proposed recognizer compares only the selected key frames of the input video sequence with the existing key frames in the prototype FSMs. This greatly enhances the recognition speed. The proposed scheme is unlike other template matching algorithms [Cui & Weng (1997), Polana & Nelson (1994), Bobick & Davis (1996)], where an image sequence is first converted into a sequence of static shape patterns and then compared to some pre-stored action prototypes during recognition

The basic steps in the proposed algorithm for recognition of dynamic hand gestures having local motions only are –

1. VOP generation from the input gesture sequence in which the gesturing hand is considered as the video object. The idea of VOP extraction comes from the MPEG-4 video coding standard, but with slight modification so as to make it suitable for the present purpose, as will be explained later in this chapter.
2. Extraction of key VOPs in the gesture sequence on the basis of object shape similarity between successive VOPs, measured in terms of the Hausdorff distance measure.
3. FSM representation of the gesture sequence with each key VOP representing a state defined in terms of the hand shape in that frame.
4. Recognition via the FSM of an input gesture, and subsequent FSM matching.

These steps are described in detail in the sections to follow.

2.2 Video Object Plane Generation

2.2.1 Overview of MPEG - 4 video coding standard

The MPEG-4 video coding standard enables various content-based functionalities for multimedia applications. Unlike its predecessors, *viz.*, MPEG-1 and MPEG-2, MPEG-4 targets more than just coding gains [Tsaig & Averbuch (2002)]. To provide new functionalities for

TH-1870_02610203

multimedia applications, such as content-based interactivity and content-based scalability, it introduces a content-based representation scheme in which a video scene is treated as a composition of different moving objects (video objects) that may be separately encoded and decoded. For this, a prior decomposition of a scene into video objects is necessary. Subsequently, each video object in a frame is represented by a video object plane (VOP) which is nothing but a snapshot of the object at that particular frame instance. Thus, each frame of a video sequence is composed of a set of VOPs corresponding to all the video objects in the scene.

A video object (VO) extraction scheme for real-time object-based applications should meet the following criteria:

1. Segmented objects should conform to human perception, *i.e.*, semantically meaningful objects should only be segmented;
2. Segmentation algorithm should be efficient and fast; and
3. Initialization should be simple and easy for users to operate, *i.e.*, human intervention should be minimized.

In view of this, the following two well established methods may be employed for MPEG-4 based video object segmentation.

2.2.2 Edge change detection for VOP generation

One approach to VOP generation is via inter-frame change detection for automatic spotting of objects and new appearances in a video sequence [Kim & Hwang (2002)]. This method of VOP generation allows for large nonrigid motions compared to the other approach based on object tracking. However, in algorithms based on inter-frame change detection, object boundaries tend to be irregular in some critical image areas due to lack of spatial edge information. This drawback may be overcome by using spatial edge information to smooth and adapt the object boundaries in a post-processing stage; the spatial edge information can be incorporated in the motion detection stage to simplify the algorithm and generate noise robust results.

TH-1870_02610203

The following three steps describe the process of edge change detection for VOP extraction.

(1) *Canny edge detection*

In the first step, the edge maps of the video frames are obtained by using the Canny edge detector. Canny proposed an approach to edge detection that is optimal for edges corrupted by white noise [Canny (1986)]. The optimality of an edge detector is related to three criteria, as given below.

- The **detection** criterion expresses the fact that important edges should not be missed and that there should be no spurious response.
- The **localization** criterion says that the distance between the actual and the located position of the edges should be minimal.
- The **elimination** of multiple responses criterion requires minimal multiple responses to a single edge. This is partly covered by the first criterion, since when there are two responses to a single edge, one of them is considered as false. This third criterion solves the problem of detecting edges corrupted by noise and works against non-smooth edge operators.

(2) *Extraction of moving edge map*

As depicted in the block diagram of Figure 2.3, the moving edge ME_n of the n^{th} frame I_n is extracted based on the edge maps DE_n of the difference image $\mid I_{(n-1)} - I_n \mid$, E_n of the current frame, and E_b of the background. The moving edge model generated by edge change is given by selecting all edge pixels within a small distance T_{change} of DE_n, *i.e.*,

$$ME_n^{\text{change}} = \{e \in E_n \mid \min_{x \in DE_n} \parallel e - x \parallel \leq T_{\text{change}}\} \tag{2.1}$$

In addition to this, the moving edges of the previous frame can be referenced to detect temporarily still moving edges, *i.e.*,

$$ME_n^{\text{still}} = \{e \in E_n \mid e \notin E_b, \min_{x \in ME_{n-1}} \parallel e - x \parallel \leq T_{\text{still}}\} \tag{2.2}$$

TH-1870_02610203

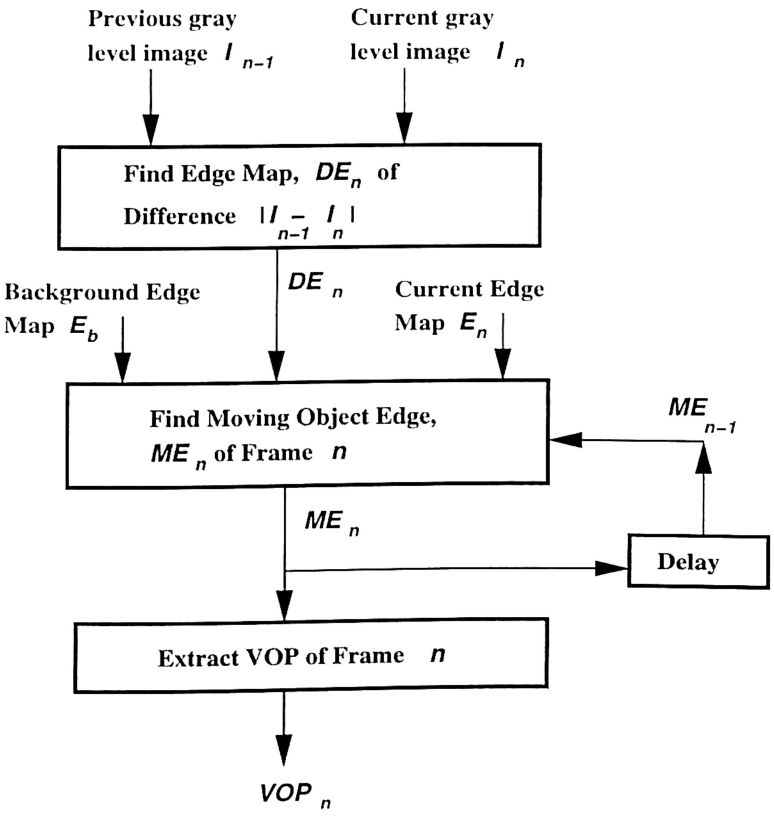


Figure 2.3: Block diagram illustrating the process of VOP extraction.

where T_{still} is some predefined constant. The final moving edge map for current frame I_n is expressed by combining the two maps as

$$ME_n = ME_n^{\text{change}} \cup ME_n^{\text{still}} \tag{2.3}$$

(3) *Extraction of VOP*

Once the edges of the moving objects in a video frame are detected from the moving edge map, the VOPs corresponding to the video objects may be conveniently extracted. The region inside the first and the last edge points in a row is the horizontal candidate for the object in a frame while that in each column is the vertical candidate. The VOP is then generated from the horizontal and the vertical candidates in a frame, and processed further by using morphological operations.

TH-1870_02610203

2.2.3 Hausdorff tracker and VOP extraction

The other method for VOP extraction is by using Hausdorff tracker for tracking moving objects over frames in a sequence. The core of this technique is an object tracker that establishes the temporal correspondence of objects throughout a video sequence. The advantage of this approach is that the tracker can keep track of objects even when they stop moving for an arbitrarily long time. This is important for content-based functionalities. After a binary model for an object of interest is derived from the edge image, the tracker matches the model against subsequent frames in the sequence and updates the model in every frame to accommodate for rotation and change in shape of the object. Thus, the output of the tracker is a sequence of binary models that guides the extraction of the VOPs [Meier & Nagan (1998)]. This is discussed in details below.

Overview of Hausdorff distance

The main issue in any pattern recognition problem is to determine the extent to which one shape is similar to another. The matching technique may be viewed as measuring the degree of resemblance between two objects that are superimposed on one another. In other words, the matching technique is usually regarded as maximization of a measure of similarity. Therefore, to recognize objects reliably, the key is to find an efficient approach for image matching leading to object search and identification. It is desired that the method is easy to implement, less time-consuming and produces reasonably good results. In model-based object recognition, the matching is accomplished through geometric comparison of shapes, and one such efficient approach is object shape matching by the Hausdorff distance measure [Huttenlocher *et al.* (1993a)]. The Hausdorff distance is a non-linear operator, which measures the mismatch between two sets of image points in space. Although the Hausdorff distance is not the only reasonable way to judge similarity between two sets of spatial points, it has been proved to be very effective in image matching problems. The Hausdorff distance measures the extent to which each point in a 'model' set lies near some point in an 'image' set and vice versa. Unlike most shape comparison methods, the Hausdorff distance is not based on finding correspondence between model and image points. It measures proximity rather than exact superposition. Thus, it is more tolerant to perturbations in the

TH-1870_02610203

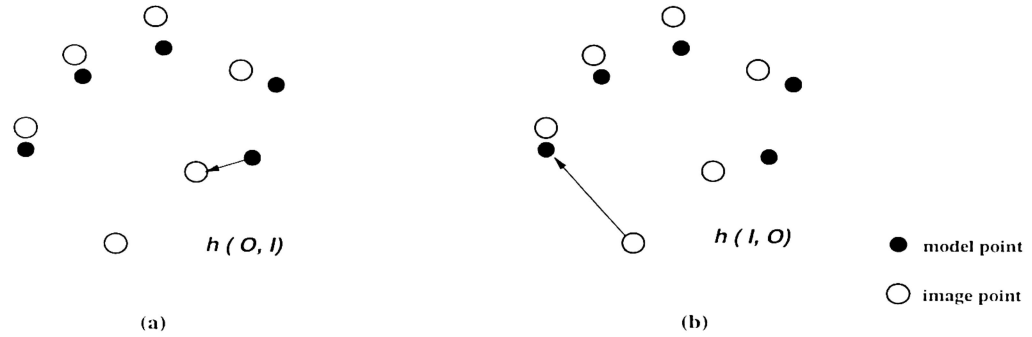


Figure 2.4: Computation of Hausdorff distance between two sets of points.

positions of the points.

As discussed earlier, the Hausdorff distance can be used to measure the similarity between two shapes. The edge pixels that form the object model are considered as a set of feature points denoted as $O = \{o_1, ..., o_m\}$, where m is the number of object edge points. The same applies to the edge image in which searching is made for the candidate object. The image set is denoted as $I = \{i_1, ..., i_n\}$, where n is the number of image edge points. The Hausdorff distance $H(O, I)$ between the two sets of points O and I is defined as given below.

$$H(O, I) = max\{h(O, I), h(I, O)\} \tag{2.4}$$

where
$$h(O, I) = max_{o_k \in O} min_{i_l \in I} \parallel o_k - i_l \parallel \tag{2.5}$$

and
$$h(I, O) = max_{i_l \in I} min_{o_k \in O} \parallel i_l - o_k \parallel \tag{2.6}$$

Here $\parallel \cdot \parallel$ is a suitable norm (*e.g.,* the Euclidean norm). Thus, for every model point o_k, the distance to the nearest image point i_l is calculated and the maximum value is assigned to $h(O, I)$. Similarly, for each image point i_l, the distance to the nearest model point o_k is computed, and $h(I, O)$ is set to the maximum over all image points. Finally, the Hausdorff distance is the larger of the two. This is illustrated in Figure 2.4. As we see, here $h(I, O) > h(O, I)$ and hence $H(O, I) = h(I, O)$. It is obvious that for $H(O, I) = d$, every model point is within the distance d from any point in I.

TH-1870_02610203

However, the definitions in equations (2.5) and (2.6) may sometime cause some problems. First, if one model or image point is outlying, the resulting Hausdorff distance will be very large, even if all other points match perfectly. Therefore, a generalized Hausdorff distance calculation, as proposed by [Huttenlocher *et al.* (1993a)], is generally used. In this, instead of using equations (2.5) and (2.6), the distances $min_{i_l \in I} \parallel o_k - i_l \parallel, \forall\ k = 1, ..., m$, are sorted in ascending order and the p^{th} value is chosen. Let this be denoted as $h_p(O, I)$. Similarly, $h_j(I, O)$ is defined as the j^{th} value of the ordered distances $min_{o_k \in O} \parallel i_l - o_k \parallel, \forall\ l = 1, ..., n$. With proper choices of p and j, we can essentially choose how many model points are near to image points and vice versa. The maximum between $h_p(O, I)$ and $h_j(I, O)$ is the generalized Hausdorff distance.

The Hausdorff distance may be used for estimating motion of an object from one frame to next by matching the object in a frame to that in the previous frame. The best match is found by minimizing the Hausdorff distance between the current frame object and the object in the previous frame translated by all possible motion vectors. The motion vector corresponding to the best match gives the motion of the object from the previous frame to the current frame. Thus, the object can be tracked over all frames in a sequence, as described below. The Hausdorff distance based matching technique is computationally cheaper and faster compared to most other object matching techniques. Moreover, it is robust to noise and changes in shape [Meier & Nagan (1998)]. Further simplification in Hausdorff distance calculation may be achieved by using distance transform algorithm, as discussed in Appendix A.

Hausdorff object tracker

For tracking of non-rigid moving objects in complex scenes, Huttenlocher *et al.* (1993b) used a model-based method in which two-dimensional models are extracted from the image data, and matched to successive frames in the image sequence. Since a model-based approach is able to exploit global attributes of the tracked object, it can provide significant advantages over purely local methods in situations where the environment is cluttered, there are multiple moving objects, or when there is a large motion of an object from one frame to the next. Moreover, the model of an object is acquired dynamically from the

TH-1870_02610203

image sequence, rather than being provided *a priori*. The main condition for successful tracking is that the two-dimensional shape of an object does not change by a very large amount between two successive frames in an image sequence. In this algorithm, an object is represented as a sequence of binary images, each corresponding to every frame in the input image sequence. Subsequently, tracking is accomplished by object matching from one frame to the next using Hausdorff distance between the corresponding binary images.

As shown in Figure 2.5, video object tracking using the Hausdorff tracker consists of two steps:

- Model initialization
- Model update

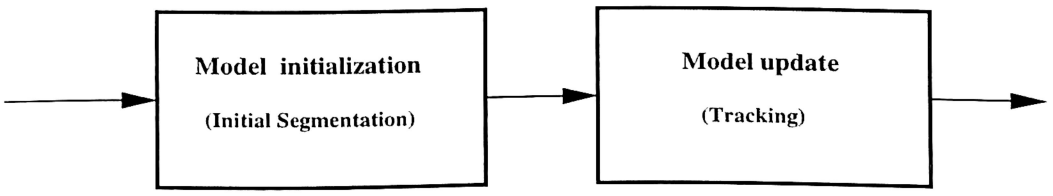


Figure 2.5: The basic two steps in a Hausdorff tracker.

The basic steps involved in model initialization are shown in Figure 2.6. The initial object model is obtained from the difference image of the first two consecutive frames and the change detection mask (CDM) is obtained by simple thresholding. The algorithm preserves the connectedness of the object model during the thinning process, which is essential for obtaining a 'nice' model. Generally, a short edge component is considered to have been generated by noise. So, during the process of connected component labelling (CCL), the final model is obtained by eliminating short edge components with an appropriate choice of threshold for the edge length.

As the tracked object moves across a video sequence, it may rotate or change its shape. To allow for this, the model is updated in every frame. As shown in Figure 2.7, tracking progresses by comparing the model M_t in a given frame I_t to the image I_{t+1} in the next frame, thereby finding the transformation specifying the best match of that model to that image. Then a new model, M_{t+1}, is formed by selecting the subset of I_{t+1} that is 'near'

TH-1870_02610203

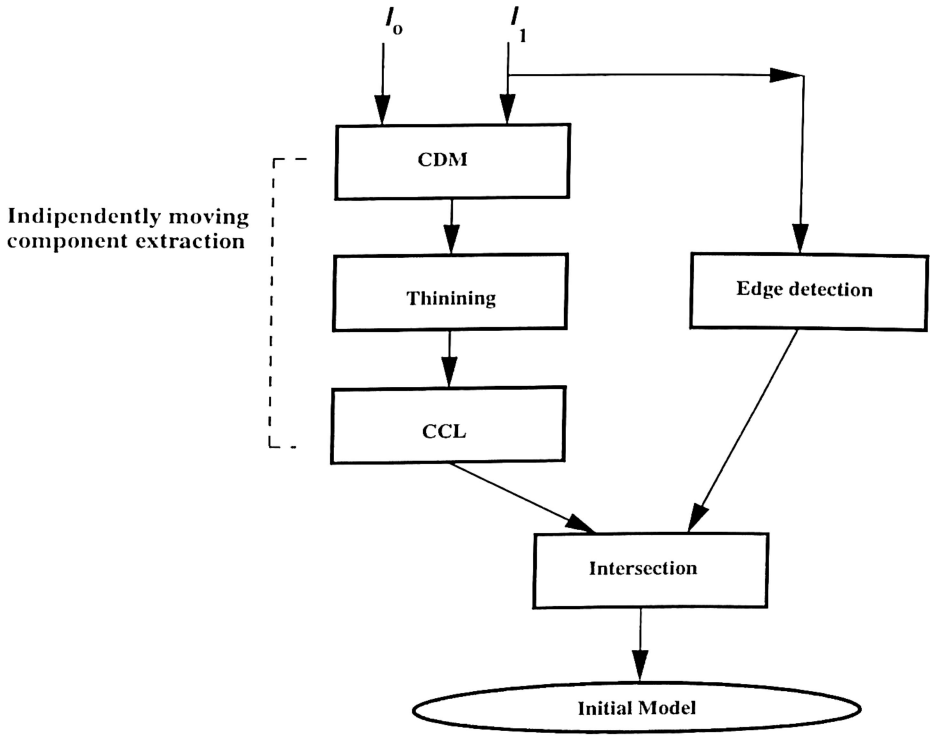


Figure 2.6: Flowchart depicting the process of model initialization in a Hausdorff tracker.

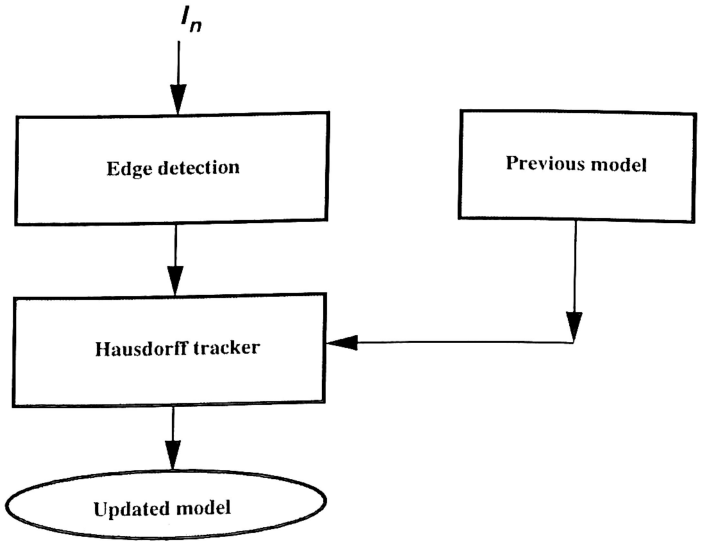


Figure 2.7: Flowchart depicting the process of model update in a Hausdorff tracker.

TH-1870_02610203

to the transformed model. This new model, M_{t+1}, represents the shape of the object in the next frame. The method of tracking an object is therefore based entirely on comparing two-dimensional geometric structures, as represented by binary image models, between successive frames in the video sequence. There is no computation of local differential image quantities such as the optical flow or motion field. Also, there is no *a priori* model of the object being tracked. The major aspects of the tracker are listed as follows.

- Decomposing the image of a moving 3D object into two parts – a 2D motion model and a 2D shape change model. The shape change is assumed to be relatively small from one frame to the next, but the motion may be relatively large.
- Capturing the 2D shape change between successive frames via analysis of 2D geometric models that evolve over time. The models provide global geometric constraints necessary for tracking an object.
- Fast 2D model matching using the minimum Hausdorff distance. The Hausdorff distance measures proximity without computing an explicit correspondence.

Extraction of VOPs

The output of the Hausdorff tracker is a sequence of binary edge images representing the object of interest. Now, the next step is to extract the corresponding object from the video sequence. This is done by finding the first and the last model points for each row as well as for each column in a frame. The pixels in between are assigned to the VOP, and thus the VOP for every frame is generated.

2.2.4 Method for VOP generation in the proposed gesture recognition scheme

The VOP generation method adopted in the proposed gesture recognition scheme may be divided into the following four stages as depicted in Figure 2.8.

- Stage 1: Initial hand model extraction.
- Stage 2: Edge detection in the input video sequence.

TH-1870_02610203

- Stage 3: Object tracking and model update.
- Stage 4: VOP extraction.

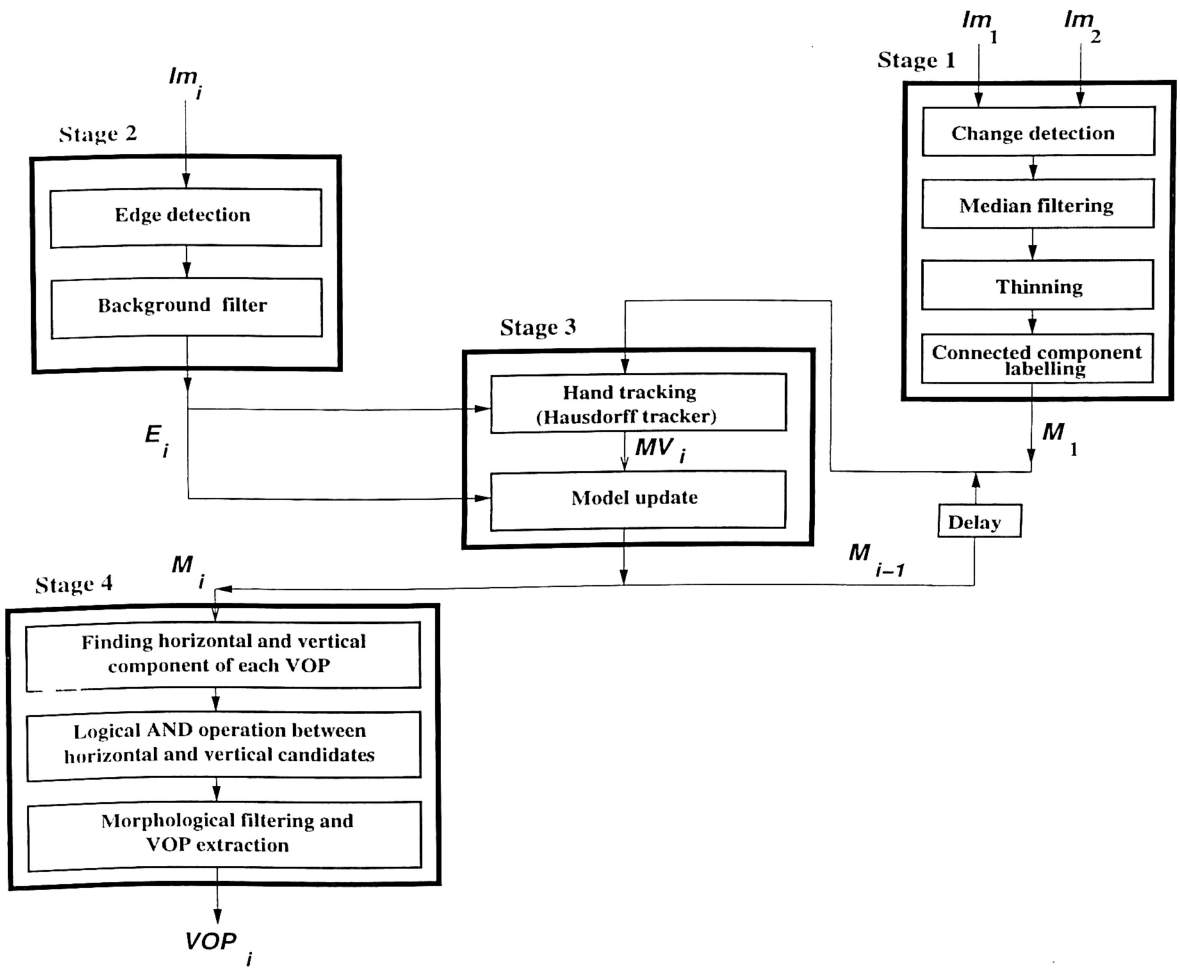


Figure 2.8: Block diagram for the VOP generation method used in our algorithm.

The RGB frames of the input gesture sequence are converted to gray scale images and the initial hand model is generated from the starting two frames, as shown in Stage 1 block of Figure 2.8. In Stage 2, the edge image for each frame is obtained. This edge information is used for model update as well as for tracking in Stage 3. Finally, in Stage 4, VOPs are extracted from the corresponding updated model.

The main tool used in this algorithm is an object tracker that matches a two-dimensional (2D) binary model of the video object against subsequent frames using the Hausdorff distance. The best match found indicates the translation that the object has undergone, and the

TH-1870_02610203

model is updated in every frame to accommodate for rotation and change in hand shape. However, the method will be effective only if the video object shape changes slowly from one frame to the next which we assume to be true in the rest of the thesis.

The video object extraction and VOP generation method was applied for object based video coding as in MPEG-4 [Meier & Nagan (1998)], object based video abstraction for surveillance [Kim & Hwang (2002)], etc. However, our algorithm differs from the method in [Meier & Nagan (1998)] and [Kim & Hwang (2002)] in the sense that we propose to use *median filtering* and *logical AND operation* suiting the purpose of application, as explained below.

Initial hand model generation

An initial model is necessary for tracking of video object in successive video frames of the input video sequence. This is accomplished via the following steps.

1. *Change detection*: This step generates a change detection mask by thresholding the difference image formed from the two starting frames in the input gesture sequence.
2. *Median filtering*: The threshold difference image contains isolated white spots. Median filter reduces these noisy spots thereby generating a smoother initial model.
3. *Thinning*: Morphological thinning is applied to reduce the width of the edges of the hand model.
4. *Connected component labelling*: This process eliminates short edges in the model.

An example of initial model generation using all these steps is illustrated in Figure 2.9.

Edge detection and background filtering

Once the initial hand model is obtained, the next task is to determine any change in the hand shape in subsequent frames with respect to the initial model. Background filtering is one desired step when VOPs are extracted from cluttered background. Removal of background edges reduces the number of points to be considered during the process of shape change detection and thus speeds up the process of model updation and tracking. However, the

TH-1870_02610203

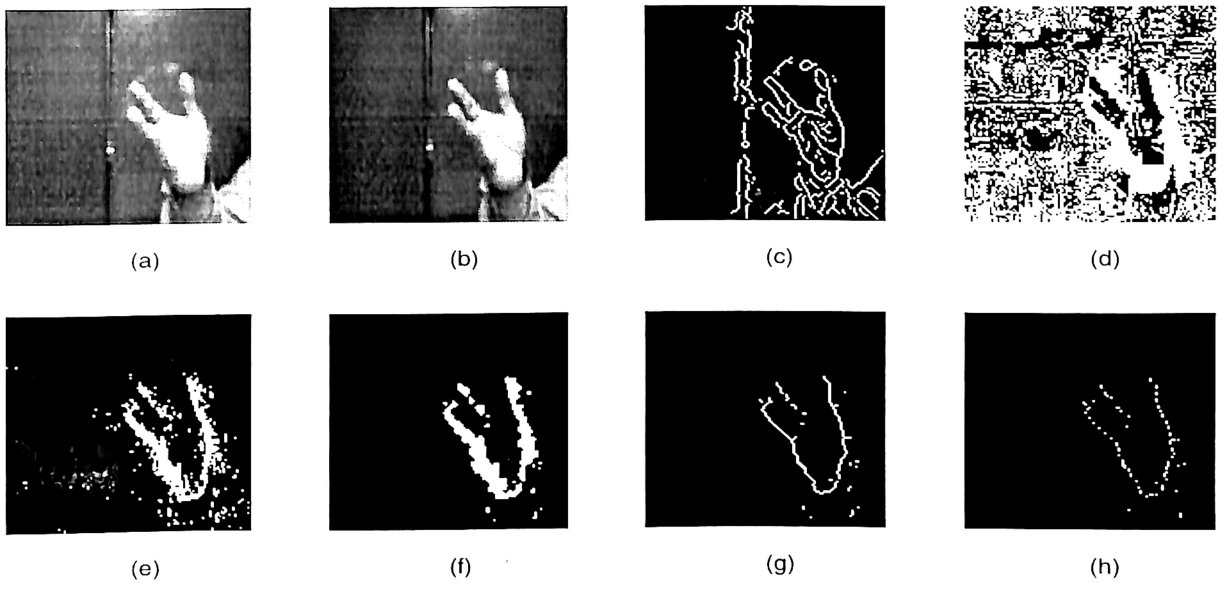


Figure 2.9: Initial model generation, (a)-(b) First two frames (c) Edge image of the first frame (d) Difference image (e) Thresholded difference image (f) Median filtering of threshold difference image (g) Thinning (h) Connected component labelling. The edge image of the first frame is included to show the background edges.

algorithm is capable of functioning even without background removal as long as the background does not change significantly. This is because the proposed algorithm uses shape variation information for gesture modeling. As long as the background edges are more or less stationary in a sequence, the performance of the algorithm is not affected.

Hand tracking and model update

The Hausdorff object tracker finds the position where the input hand model best matches the next edge image and returns the motion vector MV_i that corresponds to the best translation. As the tracked object moves across frames in a video sequence, it may rotate or change its shape. To allow for this, the model is updated in every frame using the motion vector information. First, the current model M_{i-1} is dilated and then shifted by the motion vector MV_i. Then, the portion of the edge image E_i that overlaps this modified model is selected as the new model image M_i.

For calculating the Hausdorff distance in different steps of the algorithm, we have used

TH-1870_02610203

the *distance transform algorithm* [Borgefors (1986)] described in Appendix A which in turn is computed using the Chamfer 5-7-11 mask shown in Figure 2.10. Figure 2.11 shows an example of a distance transformed image formed by computing the distance transform at all pixel points in an image.

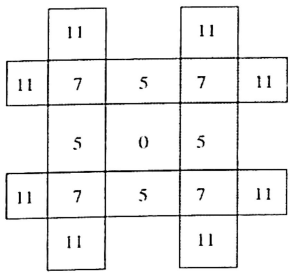


Figure 2.10: Chamfer 5-7-11 mask.

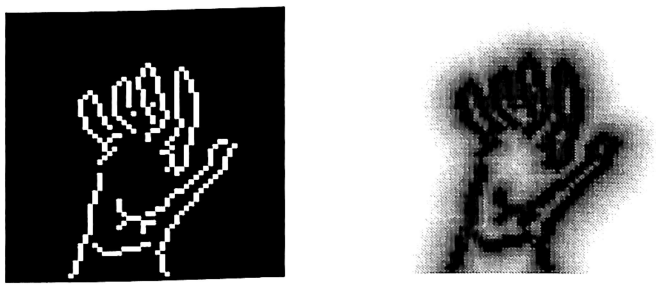


Figure 2.11: Original edge image and the corresponding distance transformed image.

Finding lost hand during tracking

The tracking method described so far may encounter tracking drift if the hand temporarily disappears from view or changes shape suddenly from one frame to the next. In such cases, no matching for the current model M_i can be found in the frame I_{i+1}. When this happens, we compare additional models to frame I_{i+1}. These additional models are the models from few previous frames, which are chosen as "canonical" views of the hand; the set of canonical hand models is constructed by selecting "distinctive looking" hand models

TH-1870_02610203

from certain frames of the image sequence. This is nothing but a "hunt down" approach, because the past history of the system is used to hunt for the missing hand. In this, the "canonical" hand models are matched to the image I_{i+1}, until either a good match is found, or the set of canonical models is exhausted. The first canonical hand model that matches the image is then used to construct M_{i+1} in the normal manner. If none of the canonical models match I_{i+1} then the tracker has lost the object (hand) in that time frame and no hand model M_{i+1} is constructed for such a frame. After each model M_{i+1} is created, if at all the hand is not lost, it is compared with the current set of canonical hand models. If the new model is sufficiently different from all the other hand models in the canonical model set, then it forms a new view of the hand and is added to the set.

The technique discussed above is very useful when the hand suddenly disappears from the capture range of the camera and reappears after few frames. Moreover, this technique can also cope with the problem of sudden change in hand shape to some extent.

VOP extraction

Finally, the VOP corresponding to the gesturing hand is generated for every frame in the input gesture sequence. After finding all the *horizontal candidate* and the *vertical candidate* for the video object (hand) in a frame, the VOP is generated by the logical AND operation between them and further processed by morphological closing and filling operations. To find the horizontal candidate, the hand model image is scanned from left to right. The edge points and all the pixels between two edge points along each scan-line are the elements of the horizontal candidate. For determining the vertical candidate, the same procedure is repeated for each column. It is to be noted that in contrary to the existing VOP generation algorithm given in [Meier & Nagan (1998)] wherein logical OR operation is used, we propose the logical AND operation between the horizontal candidate and vertical candidate[1]. This is because, by using the OR operation it is not possible to obtain a hand image in which the fingers are distinctly separated from each other which otherwise is possible with the AND operation. This is illustrated in Figure 2.12. It is seen that for hand images, logical OR operation between the vertical candidate and horizontal candidate results in a

[1]This work has been published at *ICSCI 2005* (Refer item 10 in Page 199 for details)

TH-1870_02610203

blob with all fingers joined together. This results in loss of finger shape information in the hand image. But, in our case, shape of each individual finger is important as different gestures are expressed by moving fingers in different ways. On the other hand, as shown in Figure 2.12(e), the proposed AND operation can result in a hand image in which the fingers are distinguishable as desired for the present purpose. An example of VOP generation using the proposed scheme is given in Figure 2.13 that shows the extracted VOPs from a long gesture video sequence.

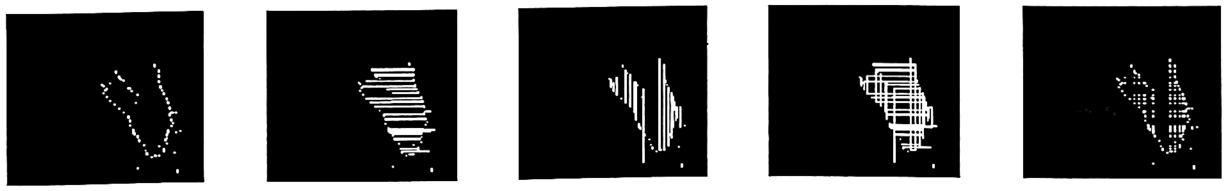


Figure 2.12: Results of logical OR and AND operations between horizontal and vertical VOP candidates. (a) Updated model (b) Horizontal candidate (c) Vertical candidate (d) Result of OR operation, where fingers of the hand are lost (e) Result of AND operation, where fingers of the hand are retained.

2.3 Key VOP Identification and Extraction

Key frame based visual representation has been used in frame-based video indexing systems where the key frames are used to represent the salient content of a video sequence. Besides visual summarization, representation using key frames allows some of the still image features (such as shape, texture and colour) to be used for video retrieval. Accordingly, many algorithms have been proposed for key frame selection in a digital video indexing and retrieval system. Some of these algorithms are applied to uncompressed video, and they involve comparing colour and motion histograms, computing pixel differences, or edge tracking [Boreczky & Lawrence (1996), Lupatini *et al.* (1998)].

While key frames provide a summary of the video content, they are unable to provide an accurate description of the individual objects in a video scene. In MPEG-4 video cod-

TH-1870_02610203

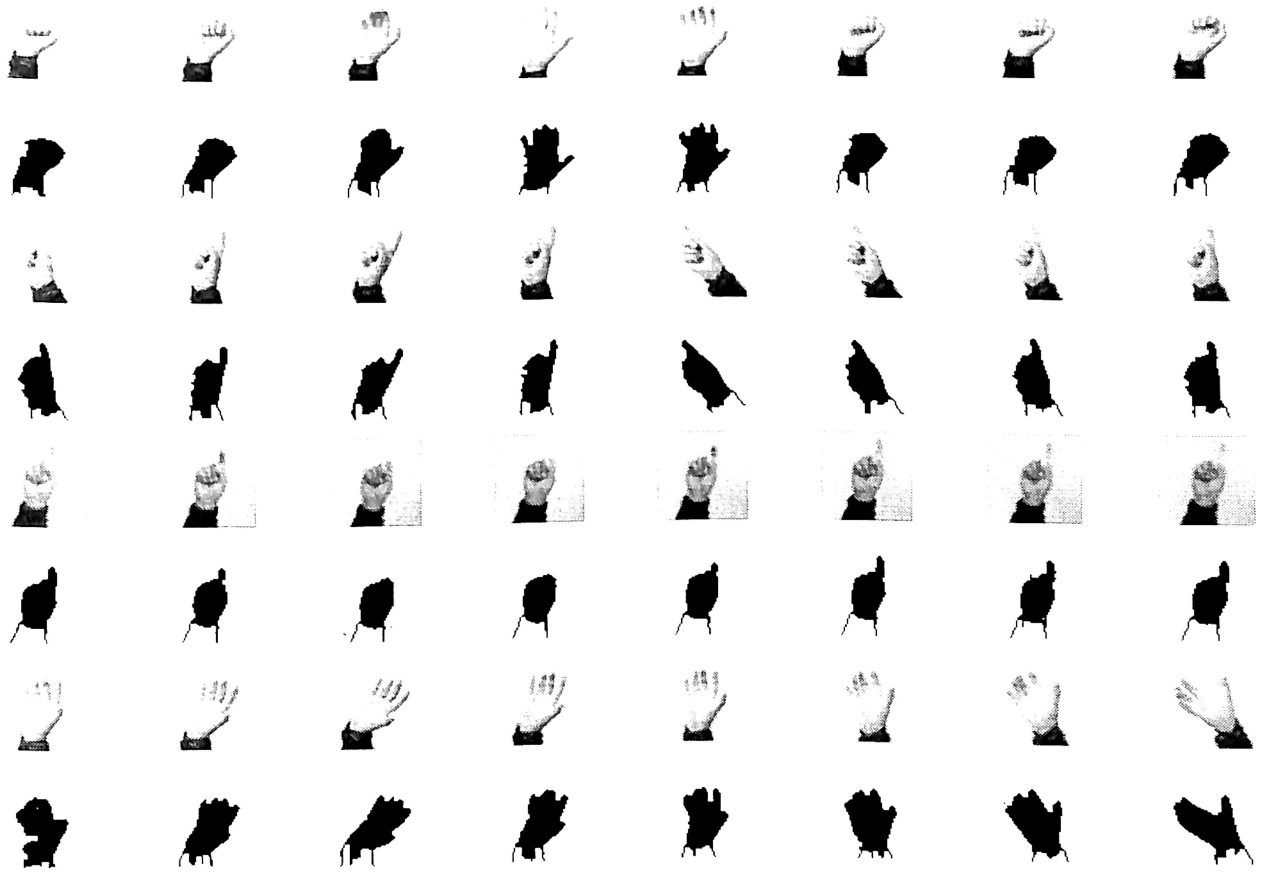


Figure 2.13: Extracted VOPs from the hand video sequences

ing standard, object-based video representation using video object planes (VOPs) offers a framework that is well suited for object-based video indexing and retrieval. In such a framework, the concept of "key frame" is replaced by that of *key video object plane (key VOP)* that may be used for visual summarization of the content corresponding to a video object. However, unlike key frame selection, very little work has been reported on key VOP selection. One of the very few algorithms for key VOP selection was given by Erol and Kossentini and used in MPEG-4 based video coding [Erol & Kossentini (2000)]. Their method is based on the shape content of video objects – significant changes in shape of the video objects are detected by comparing the shape content of the VOPs on the basis of Hamming or Hausdorff distance measures. That is, for a given key VOP in the sequence,

TH-1870_02610203

the next key VOP is the one for which this shape change measure exceeds some predefined threshold.

2.3.1 Proposed method for key VOP selection

Typically, key VOPs are selected such that they reflect significant changes in the shape, colour, and texture contents of a video object. Using shape content of a video object for key VOP selection has many advantages over using colour and/or texture content. First, the texture and colour of a video object are generally consistent during the object's lifespan. On the other hand, the shape of a video object may vary significantly due to the object's movement, structure, occlusion, etc. Therefore, a significant change in the content of a video object is more likely to be detected if the object's shape is employed as a measure. Secondly, unlike the texture information, the shape information may be coded losslessly or near-losslessly. Since the human perception is very sensitive to the artifacts at the edges, shape information is often more truthfully represented than the texture information. Therefore, using shape information makes the key VOP selection algorithm significantly less dependent on the rate-quality constraints associated with coding of the video object. Using the shape of a video object instead of its colour or texture is also potentially more computationally efficient. Because of these reasons, we propose to select key VOPs in an input gesture video sequence on the basis of the shape information of the gesturing hand.

In order to detect significant changes in the shape content, a shape similarity measure is required. For this, the Hamming or the Hausdorff distance measures may be used to estimate the difference between two hand shapes, as used in [Erol & Kossentini (2000)]. The Hamming distance measures the point-by-point difference between two shapes, whereas the Hausdorff distance measures the largest distance between the contours of two hand shapes as explained in the previous section. Both of these distance measures are commonly used in shape matching, and both have different implementation and computational complexities. However, unlike the Hamming distance where all the points in the objects contribute to the measure of the distance between two video objects contained in their respective VOPs, the Hausdorff distance depends only on the two object points (one in each of the two VOPs under consideration) that have the largest distance between them. Moreover, unlike the

TH-1870_02610203

threshold used in the Hamming distance based algorithm, the threshold used in the Hausdorff distance measure for shape similarity measurement does not depend on the activity level of the video object [Erol & Kossentini (2000)]. In view of the advantages offered by the Hausdorff distance based shape similarity measure over that based on the Hamming distance, we propose to use the Hausdorff shape similarity measure for detecting key hand poses in the gesture video sequence.

In the proposed method, we first generate binary alpha planes from the extracted VOPs of the gesture sequence. A *binary alpha plane* indicates whether a pixel belongs to a VOP or not. After getting the binary alpha planes, the key VOPs are selected on the basis of shape similarity measure using Hausdorff distance. The process for key VOP selection starts with selection of the first VOP in the video sequence as the first key VOP in the sequence. Following this whenever the Hausdorff distance between the object (hand) edge in a candidate VOP and its preceding key VOP is larger than a threshold, the candidate VOP is selected as a new key VOP. The threshold is determined as

$$T = \lambda \sqrt{\min(M_1, M_2)^2 + \min(N_1, N_2)^2} \tag{2.7}$$

where λ is a predefined scale factor that is constant for all VOPs, M_1 and N_1 are the widths and heights of the key VOP respectively, and M_2 and N_2 are the widths and heights of the key VOP candidate respectively. Thus, the threshold varies with the dimensions of the VOPs.

2.3.2 Key VOP selection using ART shape description

Another approach to shape similarity measurement is by using *Angular Radial Transform* (ART) shape descriptor that may be used for key VOP selection in a sequence of dynamic hand gesture[1]. Region-based shape descriptor is useful in expressing pixel distribution in a 2D object region. It can describe complex objects consisting of multiple disconnected regions as well as simple objects with or without holes. An ART based shape descriptor has been adopted in MPEG-7 [Erol & Kossentini (2001)]. Conceptually, the descriptor

[1]This work has been published in *ICVGIP 2004* (Refer item 4 in Page 198 for details)

TH-1870_02610203

works by decomposing the shape into a number of orthogonal 2D complex basis functions, defined by the Angular Radial Transform [Bober (2001), Manjunath *et al.* (2002)]. Consequently, the normalized and quantized magnitudes of the ART coefficients are used to describe the shape. ART is a unitary transform defined on a unit disk that consists of the complete orthonormal sinusoidal basis functions in polar coordinates. From each shape, a set of ART coefficients F_{nm} is extracted as

$$F_{nm} = \int_0^{2\pi} \int_0^1 V_{nm}^*(\rho, \theta), f(\rho, \theta) \rho d\rho d\theta \tag{2.8}$$

where $f(\rho, \theta)$ is an image intensity function in polar coordinates, and V_{nm} are the ART basis functions that are separable along the angular and radial directions, and are defined as follows:

$$V_{nm}(\rho, \theta) = \frac{1}{2\pi} \exp(jm\theta) R_n(\rho) \tag{2.9}$$

$$R_n(\rho) = \begin{cases} 1 & \text{if } n = 0 \\ 2\cos(\pi n \rho) & \text{if } n \neq 0 \end{cases} \tag{2.10}$$

The ART descriptor is the set of normalized magnitudes of these complex ART coefficients. Twelve angular and three radial functions are used ($n < 3, m < 12$) as in [Bober (2001)]. For scale normalization, ART coefficients are divided by the magnitude of ART coefficient of the order $n = 0, m = 0$. The ART coefficient of order $n = 0, m = 0$ is not used as a descriptor element because it is constant after normalization.

The distance (or dissimilarity) between the two shapes described by the ART descriptor is calculated using a suitable norm like the L^1 norm. Therefore, the measure of dissimilarity between d^{th} and q^{th} VOPs may be given as

$$\text{Dissimilarity} = \sum_i \left| M_d[i] - M_q[i] \right| \tag{2.11}$$

where M_d and M_q are the arrays of ART coefficients for these two VOPs.

TH-1870_02610203

2.4 Proposed Finite State Machine Representation of Dynamic Hand Gestures

The block diagram in Figure 2.14 depicts the process flow in our proposed hand gesture recognition system. First, the sequence of VOPs for different hand shapes in the gesture video are obtained using the method described in Section 2.2.4. During this phase, the Hausdorff tracker tracks the change in hand positions from one frame to the next in the input gesture video sequence. Next, the key VOPs are extracted by measuring inter-frame hand shape similarity using Hausdorff distance measure, as described in Section 2.3.1. By this, all redundant frames in the input gesture sequence are eliminated. Redundant frames are those frames between two key frames over which the hand does not move significantly. Therefore, the hand shapes in these frames are more or less similar to that in the immediately preceding key frame and hence may be discarded from consideration. Thus, the input gesture sequence is compressed (summarized) to a set of few key frames only. Following this, a finite state machine (FSM) for the input gesture is constructed in which every key frame corresponds to one gesture state. Finally, recognition is accomplished by matching the input sequence of gesture states and the temporal information with those of each prototype FSM contained in the database of the recognition system; the gesture corresponding to the matched FSM is the recognized gesture[1]. The database is constructed through training that involves formation of a set of FSMs for a given gesture vocabulary – one FSM corresponding to each gesture sign in the vocabulary. That means, the training phase allows the user to 'tell' the system in the FSM terminology what a gesture 'looks' like, while the recognition phase attempts to match an unknown gesture performed by the signer to a set of trained gestures.

However, instead of comparing the input FSM to all the prototype FSMs we propose to compare only the first state of the input to that of the prototype FSMs in the gesture vocabulary using ART shape descriptions. By doing so we can select one or few likely FSMs and discard all others from consideration in the recognition (matching) process. This greatly speeds up the recognition task.

[1]This work has been published in *ICSCI 2004* (Refer item 6 in Page 198 for details)

TH-1870_02610203

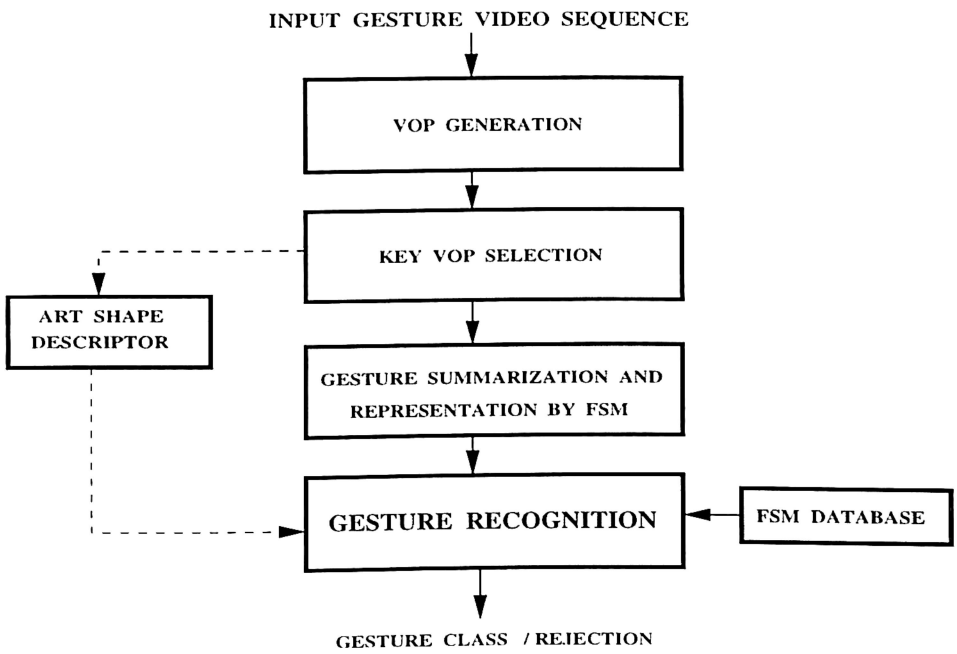


Figure 2.14: Block diagram for the proposed scheme of gesture recognition.

2.4.1 Characteristics of hand gestures

In this work, we observe several important characteristics associated with dynamic hand gesture recognition that are useful in developing the finite state machine required for recognition. These characteristic properties of dynamic hand gestures have been observed in our experiments through visual analysis of a number of gestures. These are listed below –

1. Every gesture has finite completion time, *i.e.*, a particular gesture is generally completed within a fixed interval of time.

2. During gesture recognition by a human, the human visual system generally gives more emphasis to the initial and the final states of the hand as compared to the intermediate states.

3. A human recognizes a particular gesture only when it conforms to some predefined spatio-temporal variations and crosses a minimum matching level.

Based on these observations, we develop the FSM based gesture recognition system, which covers all these aspects for recognition of hand gestures.

TH-1870_02610203

2.4.2 Proposed finite state machine (FSM) for gesture representation

Finite state machines (FSM) are used to represent different possible interaction modes and their transitions. FSMs are directed graphs that are used extensively in computer science to model a language. In gesture recognition they have been used to represent a gesture to be recognized [Davis & Shah (1994b), Hong *et al.* (2000)]. They are very easy to work with and allow recognition to be done in real-time.

An FSM consists of a finite set of states, represented as circles in most diagrams and rules for transition from one state to another. Mathematically it may be defined as

$$A = (Q, \Sigma, \sigma, q_0, F) \tag{2.12}$$

where Q is a set of states in the FSM, σ defines the set of state transitions while Σ contains the conditions for the transitions, q_0 is the initial state, and F is the set of final states. Since a gesture may be defined as an ordered sequence of states in the spatio-temporal space, in the present context, we represent a particular gesture as a sequence of key VOPs associated with their corresponding frame duration. In other words, the proposed FSM for gesture representation consists of a finite number of key VOPs as states and the corresponding key VOPs durations, as shown in Figures 2.15 and 2.16. In Figure 2.16, each state's *transition time* is shown by the symbol τ_s denoting the duration of the key VOP in terms of the number of frames between the current and the next key VOP. The transition from one state to the next takes place in a linear manner and is determined by the transition time of the key VOP. The arrow showing a self-looping of a particular state implies that it remains in the current state unless conditions for state transitions are fulfilled. The transition time τ_s is indicated along the arrow.

2.4.3 Training – forming prototype FSMs for gestures in the vocabulary

As discussed earlier, the training phase creates a prototype FSM for every gesture in the gesture vocabulary. For this, a set of training gesture sequences is collected for every gesture sign in the vocabulary. The training samples are collected from different signers performing

TH-1870_02610203

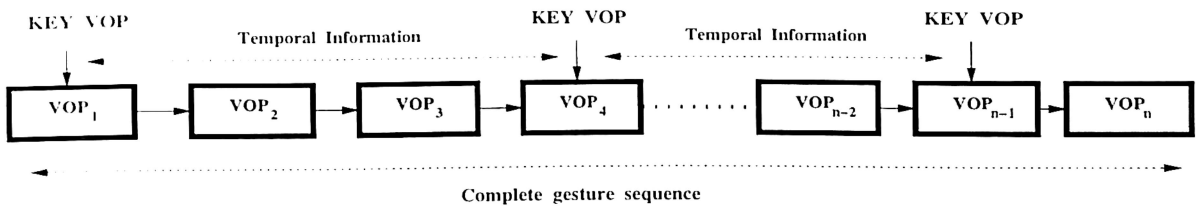


Figure 2.15: Summarization of long gesture video sequence.

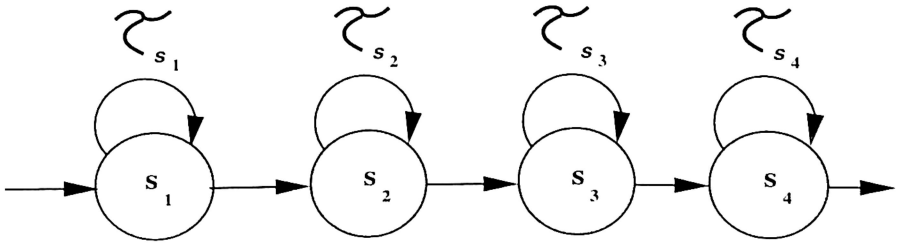


Figure 2.16: Finite state machine for gesture representation

the same gesture sign for several times. This takes into account all possible variations across different instances of one particular gesture.

As a first step towards FSM formation, VOPs are extracted and key VOPs are selected in each training sequence. The key VOPs are binarized to form binary alpha planes by assigning "**1**" to the hand-pixels in a VOP and "**0**" to the background pixels. Different training sequences for the same gesture results in different binary alpha planes for the same key VOP. The next task in system training is to group these binary alpha planes into a mean binary alpha plane that represents the state corresponding to the key VOP. Thus the states of the prototype FSM for the gesture is obtained.

Next, the duration of each key frame is measured in terms of the number of video frames between the concerned key VOP and the next key VOP. This gives the time duration during which the gesturing hand remains in the same state, that is the hand pose and shape do not change much during this period of time. Accordingly, the following measures for the l^{th} state duration are calculated from the set of training samples and are incorporated in the prototype FSM.

TH-1870_02610203

- τ_{s_l} gives the duration for which the hand may remain in the l^{th} state.
- $\tau_{\max_l}$ gives the maximum duration for which the hand may remain in the l^{th} state.
- $\tau_{\min_l}$ gives the minimum duration for which the hand may remain in the l^{th} state.

It is to be noted that in order to take care of the slight variation in gesturing speed from one sample instance to another, time duration of a state is not considered as one particular value, but in terms of time range which lies between τ_{min_l} and τ_{max_l}. These values are computed during training, as explained below. A state remains in the same condition *i.e.*, no state transition occurs only when $\tau_{\min_l} \leq \tau_{s_l} \leq \tau_{\max_l}$. Algorithm 2.1 gives the steps of generation of the FSM corresponding to a data sequence for a particular gesture.

Algorithm 2.1: Generation of FSM from a training data sequence

begin

read the training data sequence d_m

generate VOPs $(VOP_1, VOP_2 .... VOP_{n_{max}})$

read n_{max}

$KVOP_1 \leftarrow VOP_1$ \ assign VOP_1 as key VOP

$n \leftarrow 1$

$l \leftarrow 1$

do $n \leftarrow n + 1$

if shape of VOP_{n-1} and VOP_n are not similar

$l \leftarrow l + 1$

$KVOP_l \leftarrow VOP_n$

count and store τ_{s_l} for $KVOP_l$

until $n = n_{max}$

return KVOP, τ_s

end

TH-1870_02610203

Further, the number of key VOPs depends on the threshold used for shape similarity measurement – lower is the threshold value, more will be the number of key VOPs in a gesture sequence. Also, as gestures suffer from large shape variation from sample to sample, the number of key VOPs is generally not same in all the samples of a particular gesture. So, it is unlikely to obtain an FSM of unique number of states from each of the training samples for a given gesture even when the shape similarity threshold is fixed at a particular value. To take care of this, we select a fixed value for the threshold and propose to adopt a "majority-based" scheme. Let, the majority of the training samples produce FSMs of length K. Then, in order to take care of the variation in number of states of FSM, we select only these FSMs of length K while all other FSMs are eliminated from the training process. The final model for the given gesture is obtained from these selected FSMs only by taking average for each state and determining $\tau_{\max_l}$ and $\tau_{\min_l}$ for each state. The procedure is applied for all gestures in the vocabulary and a database of FSM models, one model for each gesture is generated. This database is used for the purpose of recognition, as described in the next section.

2.5 Gesture Recognition by FSM Matching

In the recognition phase, the user continuously performs unknown gestures that the system identifies as meaningful gestures. We assume that the starting point of the input gesture sequence is known. With this assumption, the FSM corresponding to the unknown gesture is generated. The first state of the FSM along with its transition time is compared to that of the first state in each of the FSM models stored in the database. All model FSMs whose first state matches with the computed first state and the transition time satisfies the time constraint $\tau_{\min_1} \leq \tau_{s_1} \leq \tau_{\max_1}$ are considered in the next stage of recognition, while the remaining FSMs are discarded from further consideration. The state matching is done via shape similarity measurement with certain degree of tolerance.

As the signer continues to make the gesture, the second key VOP in the sequence is obtained and the corresponding state and the time durations are extracted. The second state and the corresponding transition time are compared with those of the short-listed model

TH-1870_02610203

FSMs in the first step in the same way as done for the first state. Based on the comparison results, some FSMs may be discarded from the list and some will be retained. The process continues as all subsequent key VOPs in the input gesture sequence are obtained. Thus, as the gesturing progresses, the search domain gradually reduces. Finally, the input is recognized as that gesture corresponding to the FSM in the list whose final state is reached successfully.

Variation in number of FSM states during recognition

The number of key VOPs and hence the number of states in a test gesture sample will depend on the choice of threshold for key VOP selection. Also, due to hand shape variation the number of states in a gesture generally differs from sample to sample. Hence, the number of states in a test gesture sequence may not be same as that of the prototype FSM corresponding to the gesture to which the input really matches. As shown in Figure 2.17, Case I is the ideal situation, where the number of states in the two sequences are equal. For recognition of this type of gesture sequence no extra decision and/or computation is required other than state and transition time matching. For Case II and Case III, where there are missing or redundant states respectively in the test gesture sequences, straight forward state matching will not serve the purpose. Here FSM matching is performed either by skipping the redundant states in the input or by ignoring the missed states in the prototype FSM, as the case may be, while taking into account the duration of the missed or redundant states, as illustrated in the figure. For the example of missing state given in the figure, the input FSM matches to the prototype FSM if

1. the first, second and the last states of both the FSMs match perfectly satisfying the corresponding time duration constraints.
2. the third states of both the FSMs match, and
3. the time duration of the third state in the input (τ_3) satisfies the combined time constraint of the third and fourth states ($\tau_3 + \tau_4$) of the prototype FSM.

TH-1870_02610203

Similarly, for Case III in the figure, the input matches the prototype FSM if

1. the first, second, third and the last states of both the FSMs match perfectly satisfying the corresponding time duration constraints.
2. the fourth state of both the FSMs matches, and
3. the time duration of the fourth state (τ_4) in the prototype FSM satisfies the combined time constraints of the fourth and fifth states ($\tau_4 + \tau_5$) in the input FSM.

Thus, the transition time plays a crucial role in resolving the issues of missing and redundant states.

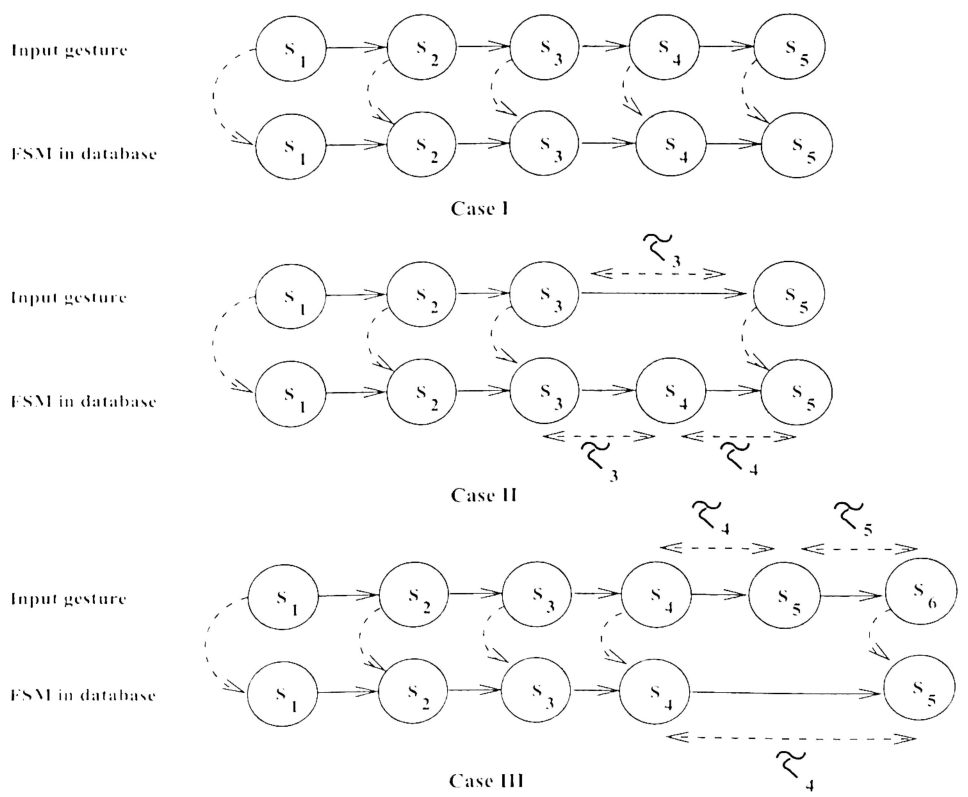


Figure 2.17: Recognition in case of FSMs having different number of states. Dotted arrows indicate the states that are compared to each other. Case I is an ideal case where the number of states in both input and database FSMs are same. Case II shows an input with one missing state while Case III shows an input with one redundant state.

TH-1870_02610203

Threshold for state matching during recognition

It is very unlikely to obtain exact shape matching between the states of the input gesture sequence and the best match FSM in the database. This is because of the spatial and shape variations that will naturally occur between different tokens of the same gesture performed at different instants of time even by the same signer. Also, there is always some variation in gesturing from one signer to another. Therefore, it is necessary to introduce some tolerance for the mismatch between the two states (one corresponding to the input and the other corresponding to the FSM to which the input is compared). The compared states are to be declared "not matching" only if the dissimilarity measure between them exceeds some predefined threshold[1]. This threshold value is decided during the training process by evaluating the maximum variation that occurs among the training samples. However, the degree of variation in hand shape is not same for all different states in a gesture and depends on the nature (hand pose) of the state. For example, less variation is observed in cases of hand completely closed or completely opened, while more variation is observed for states corresponding to half open hand. This suggests that the choice of the threshold value should be different for different states corresponding to different hand poses – more the shape variation is observed, higher the value of the threshold and vice versa. The idea behind this is that for higher variation in the shape parameter there is less chance of rejection during state matching.

Flexible state matching for recognition

Another consideration in setting the threshold for shape matching is the order in which the state occurs in a gesture sequence. We observe that generally during gesture recognition human gives more emphasis to the first and last position of the hand in the gesture sequence while the importance of all intermediate states is relatively low. Based on this observation, we design our recognizer such that it matches the shape matching criterion very strictly for the starting and ending states while it is less stringent for intermediate states, as per the *flexibility function* shown in Figure 2.18. The flexibility is introduced by weighting the state

[1]This work has been published in *IEEE Indicon 2005* (Refer item 11 in Page 199 for details)

TH-1870_02610203

matching threshold values obtained before. The modified threshold value for a particular state is given as

$$Weighted\ Threshold = Threshold \times (1 + Flexibility\ value) \tag{2.13}$$

where the flexibility value is determined by empirical study from the flexibility function given in Figure 2.18. For example, let the threshold value obtained for a particular hand shape (state) S_k be T_k as obtained through training. Then, if this state S_k appears as the first state in an FSM, then its threshold value will not change. On the other hand, if this state appears in the middle of a gesture sequence then the corresponding threshold will be modified to $1.6\ T_k$, thus allowing for more shape variation during matching.

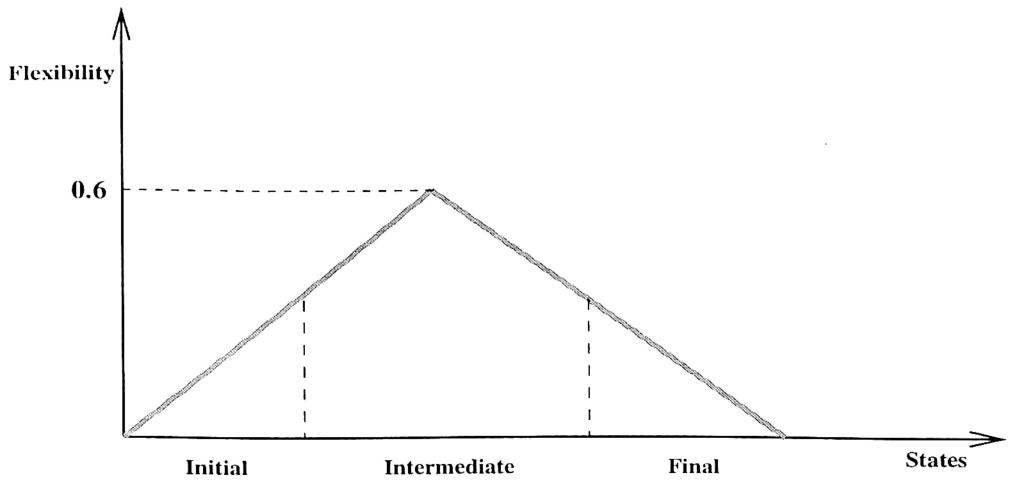


Figure 2.18: Variation of threshold flexibility for recognition.

FSM indexing for fast recognition

The searching of a suitable FSM in response to a particular input gesture may be accelerated by indexing the FSM database by using the Angular Radial Transformation (ART) defined in Section 2.3.2. For every prototype FSM in the database, the shape description of the first state is obtained using the ART. We used 35 ART coefficients for representing a shape. Therefore, for every FSM in the database we obtain a set of ART coefficients that forms a shape description vector corresponding to the hand shape in the very first state of that FSM. The shape description vectors thus obtained from the total database are then vector quan-

TH-1870_02610203

tized to form a set of codebook vectors and the FSMs are indexed as per the quantization indices in the codebook. That means, FSMs with similar starting states are assigned the same index and each index corresponds to one or more FSMs in the database, as shown in Figure 2.19.

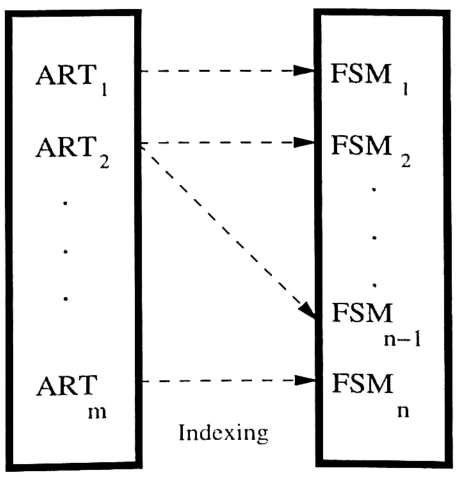


Figure 2.19: Indexing of FSMs by ART shape descriptions, where $(n \geq m)$.

During recognition, the ART shape description coefficients for the hand shape in the first state of the input gesture is computed. This set of ART coefficients is then compared (in terms of L^1 norm) to the codebook vectors generated above and the index corresponding to the best match is determined. Subsequently, prototype FSMs bearing this best match index are only considered while all other unlikely FSMs are discarded from further consideration. This greatly helps in reducing the number of searches during the recognition phase.

Recognition of static hand gestures

The FSM based representation and recognition scheme proposed for dynamic hand gestures is also extendable to static hand gestures. A static hand gesture may be viewed as a special case of dynamic hand gestures that consists of only one state occurring for indefinite time. Accordingly, a static hand gesture may be represented by an FSM that consists of only one state occurring for infinite time. It may be recognized by simple shape matching.

TH-1870_02610203

Movement epenthesis and co-articulation detection during recognition

Co-articulation is observed in the case of a continuous hand gesture sequence where different gestures are connected one after another in sequence [Shamaie *et al.* (2001)]. Movement epenthesis occurs due to unwanted hand movement during transition from one gesture to the next. As explained earlier, for recognition, the input gesture states are matched with the states of each FSM in the gesture vocabulary database obtained through training. Recognition is nothing but a string matching between the input state sequence and the state sequence of an FSM. For an input sequence, the gesture recognizer decides whether to stay at the current state or to jump to the next state based on the shape similarity and duration criteria. If all the states of the input are passed successfully then a gesture is recognized. Else, the input is considered to be a non-gestural sequence and is possibly the co-articulation phase in between two gesture phases. Thus, the proposed FSM based technique has the potential for co-articulation detection in a multi-gesture situation. The details of FSM based co-articulation detection is investigated in Chapter 5.

2.6 Experimental Results

2.6.1 Test gesture sequences

The input images were captured by a CCD camera at a resolution of 120×160 pixels. The videos were acquired as RGB colour video and converted to gray-scale frames for further processing.

For experiments, we created a video database of ten different gestures, each having local motions only. The proposed algorithm was applied to a database containing 2000 gestures of these classes performed by 10 persons. Each person was asked to do his/her gestures 20 times for each of the 10 gestures. The gesture data was partitioned into two sets one for training and the other for testing. The choice of these gestures is in view of their practical applications in human-computer interactive systems like remote robot control and hand gesture based menu activation in window based softwares. The first four gestures are selected on the basis of Sebastien Marcel's online database and other six are defined to meet

TH-1870_02610203

the above requirements. These gesture sequences are –

1. Stop-grasp-OK sequence (**OK**): Starting from a closed position, the hand (palm) gradually opens completely and then closes again.
2. Clic sequence (**CLIC**): The gesture starts with the index finger stretched upward while all other fingers are closed. The index finger then gradually closes and opens back to the initial position.
3. No sequence (**NO**): At start, the hand shows one by one finger in the up position with left sided inclination. The hand then gradually goes to the right inclination and finally to the left original inclination while making the same sign.
4. Rotate sequence (**ROTATE**):The hand is completely opened with all fingers stretched up that sways from left to right and then back to left.
5. Close to Open (**CO**): The hand just goes from closed to complete open position. This is shown in Figure 2.20 (a)
6. Open to Close (**OC**): It is just the opposite of "Close to Open sequence", as depicted in Figure 2.20 (b).
7. Close to One (**CONE**): Here the hand is initially in closed position, and then it slowly moves to one sign indicative position with the index finger stretched upward, as shown in Figure 2.20(c).
8. Close to Two (**CTWO**): This is similar to "Close to One" sequence except that the hand moves to make sign for two with index and middle fingers stretched upward. This is illustrated in Figure 2.20(d).
9. Close to One and then Two (**COT**): At the starting the hand is totally closed. Then slowly it shows one indicative position and then gradually it moves to two sign indicative position. Then the hand waits in this position for a considerable period of time to indicate the end of the gesture.
10. Open to One (**OO**): The hand is completely open at start and slowly it turns into one sign indicative position.

TH-1870_02610203

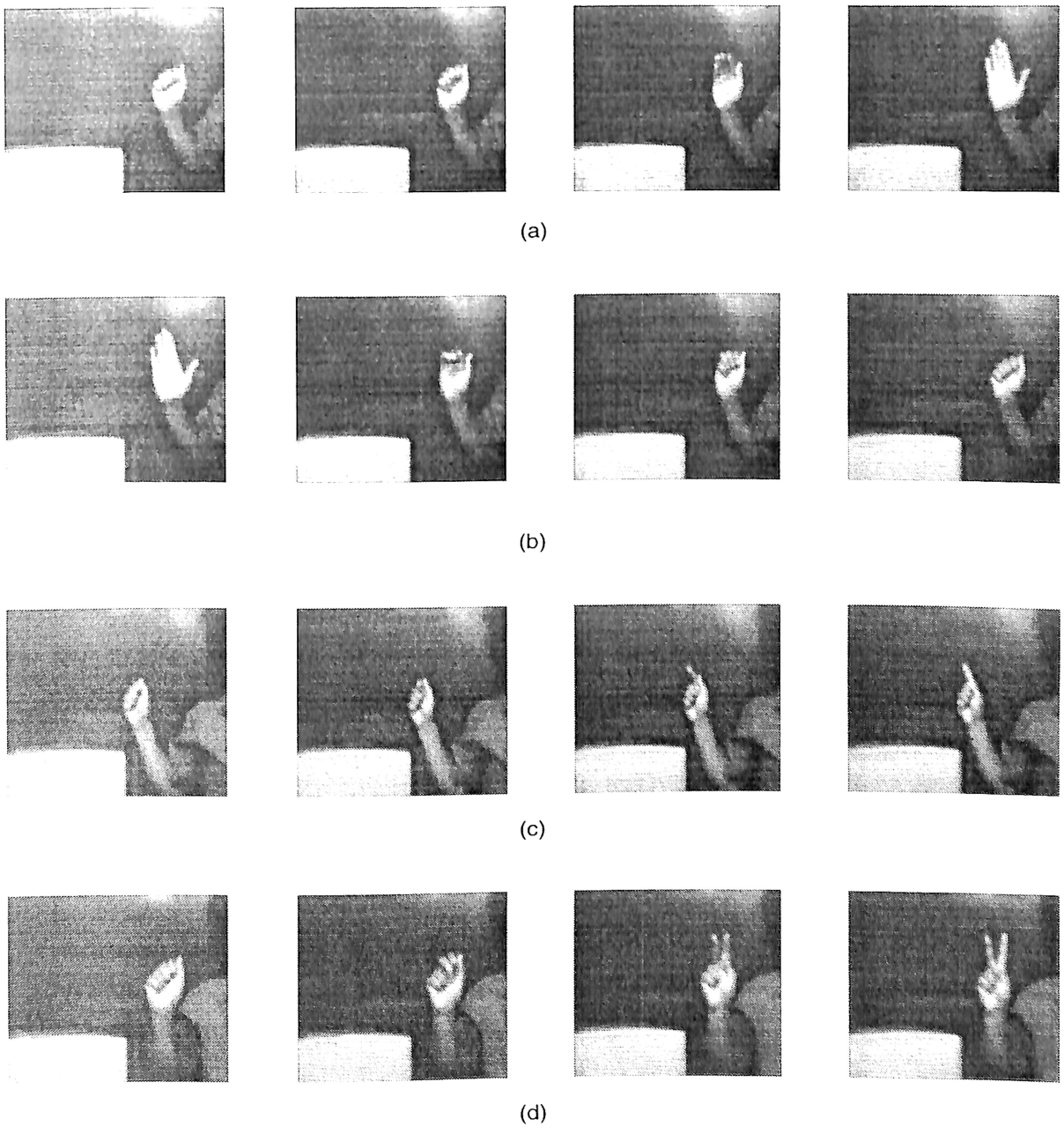


Figure 2.20: Four frames of the experimental dynamic gesture sequences (a) Close to open (b) Open to close (c) Close to one and (d) Close to two, respectively.

TH-1870_02610203

2.6.2 Experimental conditions

Experimental Constraints

In our gesture recognition experiment, the user is constrained to the following four conditions for making a gesture. Accordingly, the proposed recognizer is capable of identifying gestures only in a constrained condition. The conditions are as follows:

1. The hand is to be inserted after some time within the capture range of the camera so that the background image can be captured.
2. The hand should remain still in start position until gesture motion begins.
3. The fingers and the palm must be moved smoothly and gently.
4. After completion of a gesture, the fingers and the palm must remain in the final gesture position for a sufficient period of time to indicate the end of the gesture.

Choice of background

In our experiment, we use a comparatively simple background as seen in Figure 2.20. Moreover, all the experiments are performed under nearly constant uniform illumination condition. This is because hand segmentation and subsequent gesture recognition in a cluttered moving background and/or under variable lighting condition are beyond the scope of the current research.

2.6.3 Gesture summarization

Selection of parameters

A VOP was generated from each frame using the procedure explained in Subsection 2.2.4. The values of different parameters needed in our experiment for various stages of VOP generation are tabulated in Table 2.1. The parameters are determined by trial and errors. Some of the experimental results are presented in Figures 2.9–2.13.

TH-1870_02610203

Table 2.1: Different parameters used for VOP generation

Stages	Parameter and Value	Comment
Change detection	Threshold = 8	Determined by trial and error and then kept constant for all the sequences.
Edge detection (Canny detector)	SIGMA= 0.25 Low Threshold= 0.08 High Threshold= 0.2	Standard deviation of Gaussian and Hysterisis thresholding.
Median Filtering	Window size 3×3	Removes isolated edge points.
Connected component labelling	Edge-length Threshold = 8	Edges shorter than 8 are eliminated.
Background filtering	BGS= 1, BGE=10	Starting and Ending frame no. for background reference.
Hand tracking	$p = j = 10000$ X_SEARCH= 16 Y_SEARCH= 16	p, j represents rank corresponding to generalized Hausdorff distance. (Determined by trial and error) Object searching range.
Model update	DIST_THRES= 5	Within this distance, pixels of the old model are updated.
VOP extraction	Filter Threshold = 2	Short undesired edges are removed.

TH-1870_02610203

Key VOPs extraction

The first set of experiments illustrates the concept of key VOPs. The scale factor λ in equation 2.7 is experimentally chosen as 3 and kept constant throughout the experiments. For illustration, the image sequences corresponding to "Rotate", "No", "Clic" and "Stop-Grasp-OK" from the Sebastien Marcel's online database were considered. The results of key VOP selection for these four image sequences, are shown in Figures 2.21–2.24. The first row of each figure shows the original image sequence, second row shows the difference edge images, third and fourth rows show horizontal and vertical candidates of a particular VOP respectively. The AND combination of vertical and horizontal candidates of each VOP is shown in the fifth row. After morphological filtering we get the binary alpha plane corresponding to each VOP, as shown in the sixth row. The last row of each figure shows the finally extracted key VOPs with all in-between redundant frames discarded.

Table 2.2 shows the number of key VOPs obtained in the test video sequences. We observe that the number of key VOPs obtained is considerably less than the total number of frames in the sequence. So, gesture summarization will indeed reduce the computational burden in the recognition process.

2.6.4 Gesture classification results

The classification results on the test data using the proposed method of recognition are shown in Table 2.3 in the form of the *confusion matrices*, the accuracy rates, the error rates and the rejection rates. A confusion matrix consists of rows indicating the actual pattern (gesture) classes and columns for the classes recognized by the recognizer. Hence the scores on the leading diagonal represent correctly classified gestures. Scores off the leading diagonal illustrate how the incorrect classifications are distributed. The other terms in the table are explained in Section 1.3.1.

It is observed that good recognition performance is achieved for all the gestures under consideration. The highest recognition rate of 96.7% is achieved for the **NO** sequence and the **OC** sequence. However, we note that, the proposed recognizer makes false judgement

TH-1870_02610203

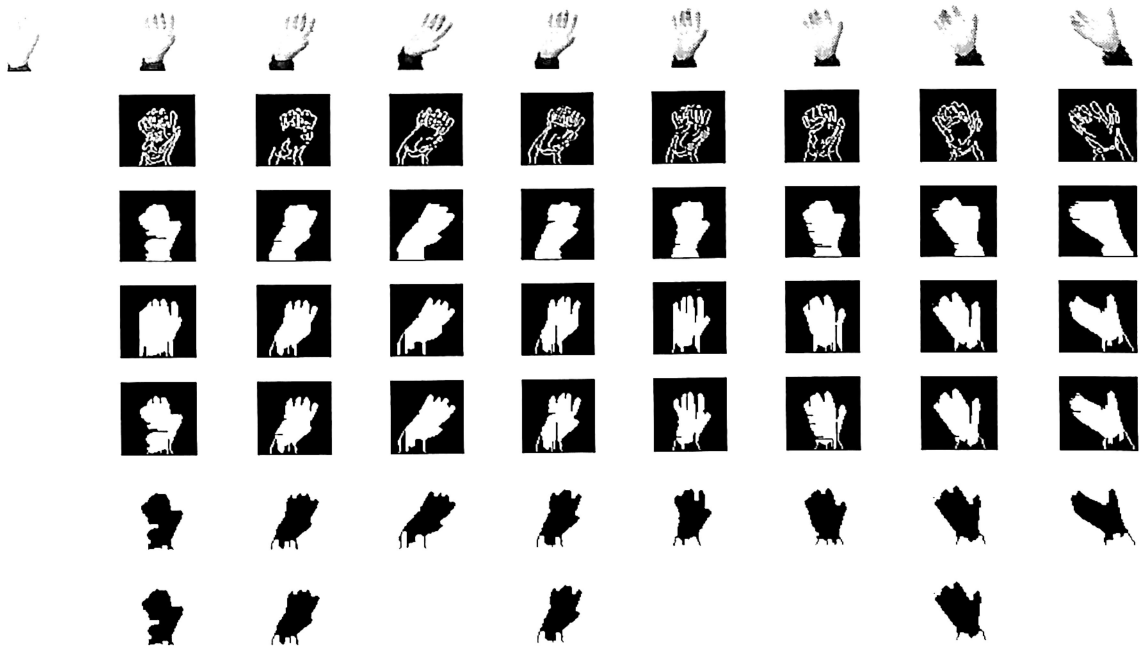


Figure 2.21: Key VOP selection for "Rotate" sequence

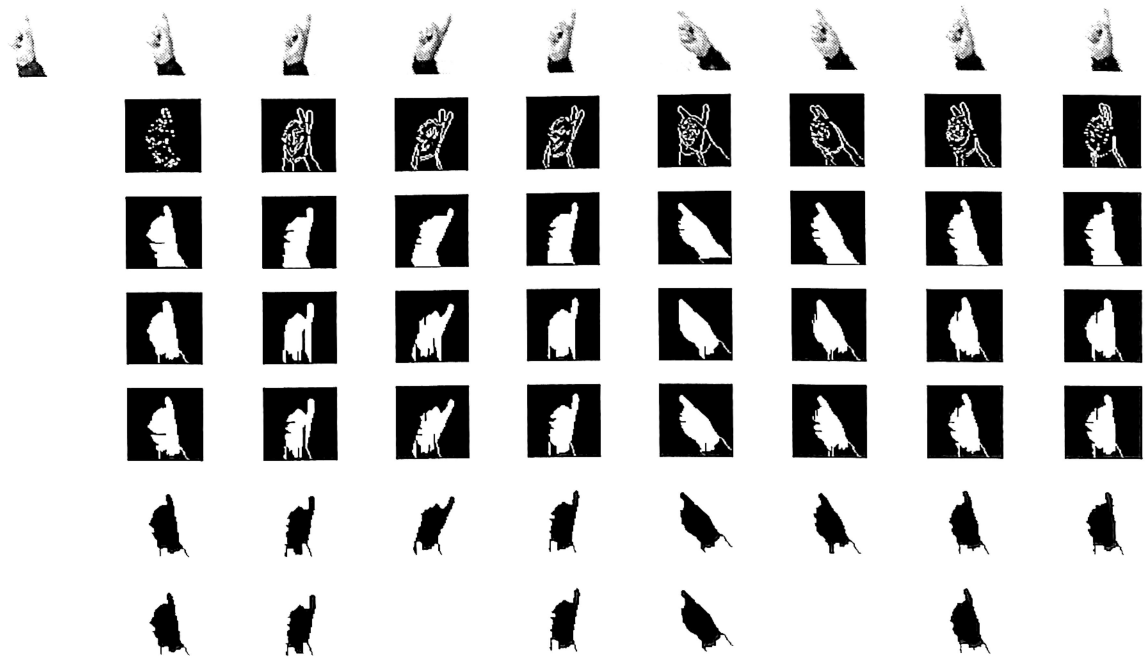


Figure 2.22: Key VOP selection for "No" sequence

TH-1870_02610203

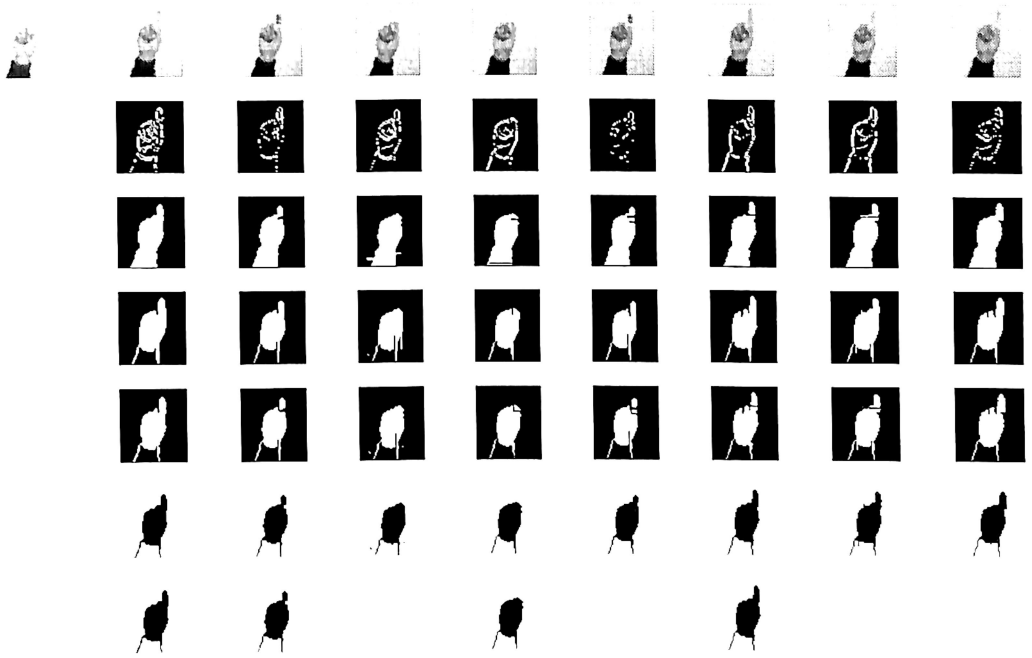


Figure 2.23: Key VOP selection for "Clic" sequence

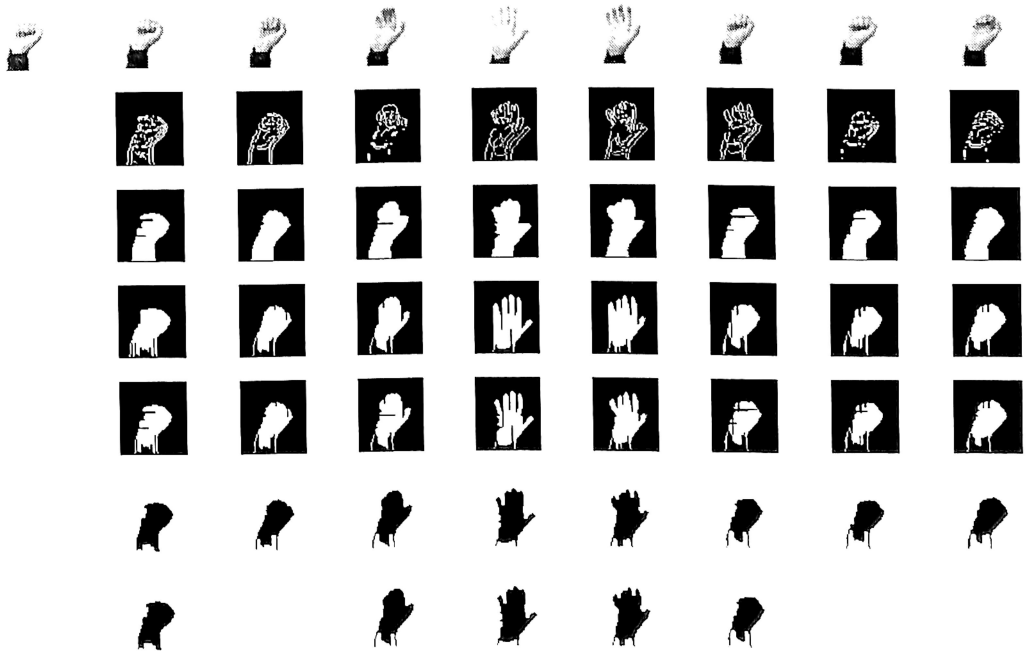


Figure 2.24: Key VOP selection for "Stop Grasp OK" sequence

TH-1870_02610203

Table 2.2: Table showing summarization of the gesture sequences

Gesture sequence	No. of frames in the sequence	Total no. of frames in summarized sequence
OK	72	6
CLIC	67	5
NO	70	5
ROTATE	79	5
CO	61	4
OC	63	4
CONE	73	5
CTWO	86	7
COT	70	5
OO	70	5

in classifying **CTWO** and **COT** sequences in some critical conditions. The ambiguity in recognition occurs if the hand makes very slow transition from one state to the next. This suggests that for the **COT** sequence, the hand should not stay in 'one' indicating position for a long time and transition from 'one' to 'two' should be very gradual and steady to avoid this ambiguity.

Results of gesture recognition on the same sequences using an HMM (hidden Markov model) classifier [Ramamoorthy *et al.* (2003)] are given in Table 2.4 for comparison. It is seen that gesture recognition and classification accuracy rate of the proposed method is comparable to that of the HMM based recognition. Similar performance of classification is observed in the case of each individual sequence.

TH-1870_02610203

Table 2.3: Experimental Results: Confusion matrix showing the gesture recognition rate using the proposed FSM-based method.

No. of training samples per class: 50

No. of test samples per class: 150

Actual class labels	No. of test pattern assigned to different classes											Acc. rate	Err. rate	Rej rate
	OK	CLIC	NO	ROTATE	CO	OC	CONE	CTWO	COT	OO	Reject			
OK	139	0	0	0	5	6	0	0	0	0	0	92.7	7.3	0.0
CLIC	0	144	0	0	0	0	6	0	0	0	0	96.0	4.0	0.0
NO	0	0	145	0	0	0	0	0	0	5	0	96.7	3.3	0.0
ROTATE	5	0	0	140	0	0	0	0	0	0	5	93.3	3.3	3.3
CO	6	0	0	0	144	0	0	0	0	0	0	96.0	4.0	0.0
OC	5	0	0	0	0	145	0	0	0	0	0	96.7	3.3	0.0
CONE	0	4	8	0	0	0	138	0	0	0	0	92.0	8.0	0.0
CTWO	0	0	0	0	0	0	0	133	11	0	6	88.7	7.3	4.0
COT	0	0	0	0	0	0	7	5	132	0	6	88.0	8.0	4.0
OO	0	0	6	0	0	0	5	0	0	139	0	92.7	7.3	0.0
Overall Average												**93.3**	**5.6**	**1.1**

TH-1870_02610203

Table 2.4: Experimental Results: Confusion matrix showing the gesture recognition rate using HMM classifier.

No. of training samples per class: 50

No. of test samples per class: 150

Actual class labels	No. of test pattern assigned to different classes											Acc. rate	Err. rate	Rej rate
	OK	CLIC	NO	ROTATE	CO	OC	CONE	CTWO	COT	OO	Reject			
OK	144	0	0	0	6	0	0	0	0	0	0	96.0	4.0	0.0
CLIC	0	139	0	0	0	0	10	1	0	0	0	92.7	7.3	0.0
NO	0	0	144	0	0	0	0	0	0	6	0	96.0	4.0	0.0
ROTATE	6	0	0	139	0	0	0	0	0	0	5	92.7	4.0	3.3
CO	5	0	0	0	145	0	0	0	0	0	0	96.7	3.3	0.0
OC	10	0	0	1	0	139	0	0	0	0	0	92.7	7.3	0.0
CONE	0	5	4	0	0	0	141	0	0	0	0	94.0	6.0	0.0
CTWO	0	0	0	0	0	0	0	132	12	0	6	88.0	8.0	4.0
COT	0	0	0	0	0	0	8	4	129	0	9	86.0	8.0	6.0
OO	0	0	7	0	0	0	6	0	0	131	6	87.3	8.7	4.0
Overall Average												**92.2**	**6.1**	**1.7**

TH-1870_02610203

Static gesture recognition

As pointed out earlier, the proposed system degenerates into shape-based matching in the case of static gestures. To test the recognition capability of the proposed shape matching in the case of static hand gestures, we used four hand shapes as shown in Figure 2.25. In all these cases, 50 training samples and 150 test samples were used. Table 2.5 shows the accuracy rates in case of the selected static gestures. The recognizer gives an average accuracy of more than 99%, which is a very good recognition rate for any Human Computer Intelligent Interaction (HCII) system. The slightly lower recognition performance may be attributed to the confusion in classification between gestures 'One' and 'Two' when two fingers are not clearly distinguishable in the later case.

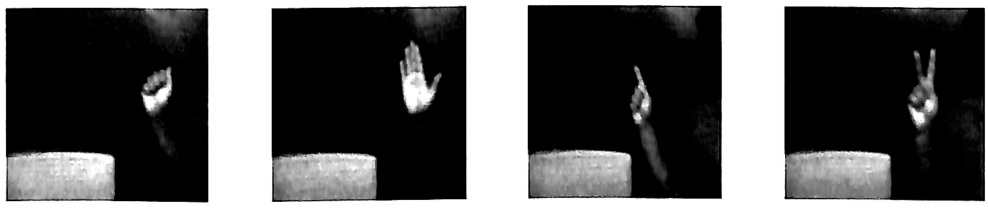


Figure 2.25: Some of the experimental static hand shapes showing "Close", "Open" "One" and "Two", respectively.

Table 2.5: Table showing static gestures recognition rates

No. of training samples per class: 50

No. of test samples per class: 150

Static gestures	Recognition rates
CLOSE	99.3
OPEN	100.0
ONE	98.7
TWO	98.0

TH-1870_02610203

2.7 Concluding Remarks

In this chapter, we have developed an FSM based novel technique for recognition of hand gestures having local motions only. One notable advantage of finite-state machine representation of gesture is that it can handle gestures of different lengths. In the gesture representation scheme, we have considered several characteristics that are observed in real gesture sequences. For recognition, we have taken clues from the gesture recognition by human and have accordingly set different matching thresholds for different states in an input gesture.

As discussed in this chapter, the Hausdorff tracker used in our method takes care of rotation and scaling of the hand thereby saving some amount of computations necessary for determining affine motions. Further, incorporation of median filtering in the model formation stage enhances the tracking performance. More importantly, introduction of logical AND operation in the final VOP generation step preserves the finger information in the gesturing hand.

Experimental results confirm that by using key VOPs, a particular gesture can be uniquely modeled in terms of a finite number of states with every key VOP as a state. The advantage of key VOP based state representation is that only the shape similarity measurement for key VOPs are required instead of all frames in the video sequence. This greatly enhances the recognition speed.

We compared our proposed recognition system to an HMM based recognition system. HMM classifiers are very efficient for classifying dynamic data patterns. Nonetheless, our system yields slightly better recognition accuracy for the selected gestures than the HMM based approach. This proves the effectiveness of the recognition system proposed in this chapter

TH-1870_02610203

Chapter 3

Recognition of Hand Gestures having Global Motions only

In this chapter, we consider another form of dynamic hand gestures that are characterized by total movement of the hand (arm) in space. For these types of gestures, the shape of the hand (palm) during gesturing does not bear any significance. We refer to this type of gestures as hand gestures having global motion only. For these types of gestures, hand motion trajectory and the associated trajectory features are useful for recognition. In our work, we propose a model-based method for tracking hand motion in space, thereby estimating the hand motion trajectory. We also employ the dynamic time warping (DTW) algorithm for time alignment and normalization of spatio-temporal variations that exist among samples belonging to the same gesture class. During training, one template trajectory and one prototype feature vector are generated for every gesture class. Features used in our work include some static and dynamic motion trajectory features. Recognition is accomplished in two stages. In the first stage, all unlikely gesture classes are eliminated by comparing the input gesture trajectory to all the template trajectories. In the next stage, feature vector extracted from the input gesture is compared to all the class prototype feature vectors using a distance classifier. Experimental results demonstrate that our proposed trajectory based gesture classifier is capable of yielding more than 95% of classification accuracy and hence is suitable for application in a Human Computer Interaction (HCI) platform.

TH-1870_02610203

3.1 Proposed Scheme for Gesture Recognition

In this chapter, we consider one form of commonly used dynamic hand gesture in which the 2D gesture trajectory alone builds up a particular message. For these types of gestures, as the hand moves in space, there is not much change in the hand shape, finger configuration and palm orientation. That means, in these types of hand gestures there is global motion of the hand as a whole, while there is no partwise local movement in the hand. Examples of few such gestures are shown in Figure 3.1.

Dynamic hand gestures of this kind may be interpreted as a set of points in a spatio-temporal space represented by $\{(x_1, y_1), (x_2, y_2), ...)\}$, where (x_i, y_i) is the i^{th} instant spatial position of a predefined reference point on the hand. This is illustrated in Figure 3.2 that shows the hand trajectory in the spatio-temporal space, where the spatial position of the hand in every frame along the temporal (time) axis is projected onto the xy-plane. Therefore, an essential step for recognizing such a gesture is to estimate the 2D gesture trajectory. Eventually, the gesture can be recognized by identifying the trajectory via analysis of some selected hand motion parameters.

As mentioned above, the first task in recognizing gestures having global motion is to track hand motion from the input gesture video sequence and subsequently estimate the trajectory along which the hand moves during gesturing. While trajectory estimation is quite simple and straightforward in glove-based hand gesture recognition system that provides spatial information directly [Quam (1990), Sturman & Zeltzer (1994)], trajectory estimation in vision-based system may require applying complex algorithms to track hand positions across frames using image silhouettes and edges. One such method is given by Yang

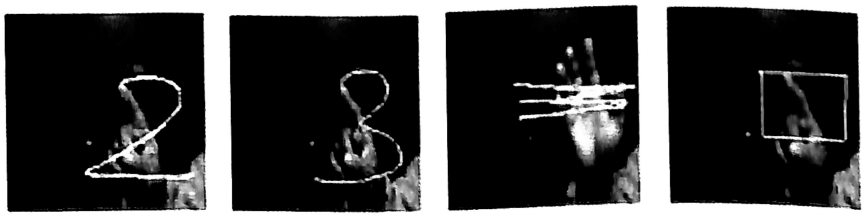


Figure 3.1: Gestures representing "two", "three", "wavy hand" and "square" respectively.

TH-1870_02610203

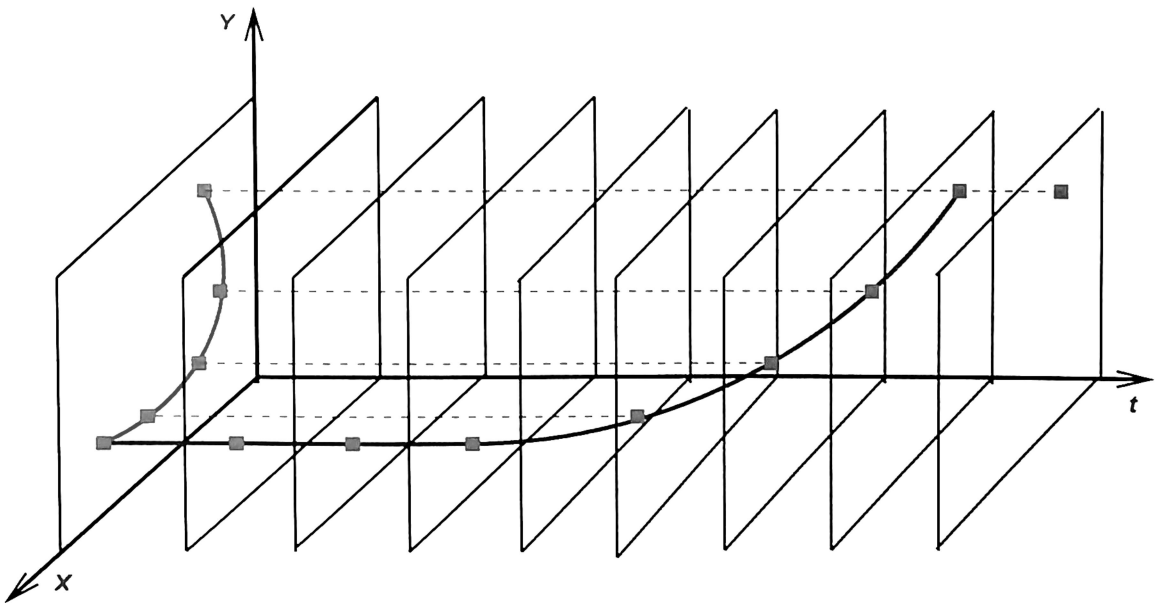


Figure 3.2: 2D hand motion trajectory in spatio-temporal space.

et al. (2002) that uses skin colour to segment out hand regions and subsequently determines 2D motion trajectory. However, hand tracking algorithms in general are plagued by some difficulties such as large variation in skin tone, unknown lighting conditions and dynamic scenes.

For simple and small motion, the Kalman filter may be employed to estimate, interpolate and predict the motion parameters. But, this requires fine initial model and also demands additional computations whenever there is any rotation and change in hand shape. Also, the Kalman tracker is very much sensitive to background noise and is based on small motion assumption that often fails to hold in hand gesture motion [Wu & Huang (2001)]. Moreover, the Gaussian assumption in case of Kalman filtering is usually invalid in real world situation.

In [Black & Jepson (1998a)], Black and Jepson extended the Condensation (conditional density propagation) algorithm to determine temporal trajectories . Since a sampling technique is used to represent the probability density in the condensation algorithm, their approach avoids some difficulties associated with Kalman filtering. Motivated by this, Black

621.3
BHU/V
826
TH-1870_02610203

THESIS
Lakshminath Bezbaroa Central Library
Indian Institute of Technology Guwahati
ACC. No. TH 1870
28/3/19

et al. used the mixed state condensation algorithm to recognize a large number of gestures using temporal trajectories [Black & Jepson (1998b)]. Here, the authors focused solely on applying condensation to identify temporal trajectories, and left for future work the problem of leveraging the temporal models as constraints on object tracking. As such, they ignored the tracking aspect of the problem all together; only simple phicons (physical object icons) of distinctive colours are tracked by a colour blob tracker. Developing temporal trajectories using the condensation algorithm is a relatively ad-hoc process that may have difficulties in capturing the way in which different individuals make the same gesture. Moreover, in this approach all the trajectory points are considered for building up a trajectory model as well as for recognition. This seems to be computationally inefficient.

In an attempt to overcome all the above mentioned difficulties, we propose a model-based method for tracking hand motion in a gesture sequence[1]. In our work, we use the concept of object-based video abstraction via segmenting the frames into video object planes (VOPs) with the hand as the video object. Next, a binary image for the moving hand is created and is used for tracking in subsequent frames using the Hausdorff tracker. The algorithm for the tracking process has been described in the preceding chapter. Next step of the proposed scheme is to select few key VOPs. Following this, we employ a centroid based method for estimating the hand trajectory, where centroids of the key VOPs are calculated on the basis of moment equations as well as from respective motion vectors. The computed centroids are then joined in the temporal sequence to obtain the final estimated trajectory. This is followed by final smoothing of the extracted trajectory and subsequently approximated the trajectory by applying MPEG-7 motion trajectory representation technique [Manjunath *et al.* (2002)].

For pattern classification purpose, we propose to compare an input trajectory pattern with all predefined prototype gesture trajectory patterns using Dynamic Time Warping (DTW). The DTW is a template-based dynamic-programming matching technique that is applied to problems with temporal variability. It provides for nonlinear time alignment and normalization thereby eliminating the temporal variability that exists between different trajectories. Further, in our proposed pattern matching technique, instead of considering all

[1]This work has been published in *ICSCI 2005* (Refer item 10 in Page 199 for details)

TH-1870_02610203

the trajectory points, we map some selected points (key points) on the template trajectories onto the test trajectory. This discrete trajectory matching technique greatly reduces the computational burden.

The next important step in gesture recognition is to select suitable and appropriate motion features that describe the extracted input trajectory in terms of numerical values. Since hand gestures generally show large variations in shape, motion and textures, selecting good features is crucial for accurate recognition. In view of the present problem, we propose to use some basic trajectory features like the trajectory length, the number of significant curves in the trajectory and the velocity features so as to accomplish trajectory guided recognition successfully. Velocity features include the mean and the standard deviation of the speed by which the hand moves during gesturing, and the number of minima in the velocity profile that corresponds to the number of sharp corners in the trajectory. While the trajectory length and number of corners may be considered as static features, velocity features are the dynamic features considered for defining the nature of the hand motion. Finally, trajectory classification is accomplished on the basis of the Euclidean distance between the set of static and dynamic features extracted from the input and each of the prototype gesture trajectories contained in the gesture vocabulary database[1].

In a nutshell, the proposed method estimates hand motion trajectory from the segmented out VOPs which preserve the spatio-temporal information about the input hand gesture. Recognition of gestures is then accomplished via matching of trajectory shape and trajectory features. For fast recognition, we select few key points on the gesture trajectory and use only these key points for trajectory matching as well as for feature extraction. Key-point based trajectory representation significantly reduces memory requirement for storing the trajectory information. In the proposed method, spatio-temporal variation is collectively coped with the DTW-based nonlinear time alignment of trajectories and the maximum boundary deviation (*MBD*) - based shape similarity measurement. Once a particular gesture has been recognized, it may be mapped to a corresponding action for human-computer interaction. Thus the proposed system is suitable for HCI applications.

[1]This work has been published in *Journal of Experimental and Theoretical Artificial Intelligence* (Refer item 1 in Page 198 for details)

TH-1870_02610203

3.2 Proposed Trajectory Estimation Algorithm

3.2.1 Hand tracking and estimation of the motion vector

One obvious and essential step in hand motion trajectory estimation is to determine the motion of the gesturing hand from one frame to the next in the sequence. For this, we employ a hand tracking algorithm that is based on an object tracker that matches a two-dimensional binary image of the object (hand) in subsequent frames using the Hausdorff distance. In our proposed technique, VOPs for different hand positions are obtained from the input gesture video sequence, where the hand is considered as the video object. The details of VOP estimation are described in Chapter 2. In the next step, the Hausdorff object tracker finds the position where the input hand image best matches the next edge image and returns the motion vector that represents the best translation. The best match found indicates the translation the object has undergone, and the model is updated with every frame to accommodate for rotation and change in shape.

In the hand tracking process, we use a simplified Hausdorff tracker with reduced computational load. Considering the VOPs O and I, our aim is to find the motion vector between O and I through the Hausdorff tracker. As explained in Chapter 2, an effective way to approximate the Hausdorff distance is the generalized Hausdorff distance in which the distances $min_{i_l \in I} \parallel o_k - i_l \parallel$ for all $o_k \in O$ are sorted in ascending order and the p^{th} value is chosen as $h_p(O, I)$. Similarly, $h_j(I, O)$ is defined as the j^{th} value of the ordered distances $min_{o_k \in O} \parallel i_l - o_k \parallel$ for all $i_l \in I$. The maximum of these two gives the generalized Hausdorff distance $H(O, I)$. In order to speed up the computation, the generalized Hausdorff distance is calculated via distance transform [Borgefors (1986)] which in turn is accomplished using the Chamfer $5 - 7 - 11$ mask. Further reduction in computation is achieved by searching the shifted hand image only at pixel locations in a search window around the model image in subsequent VOPs. The algorithm for determination of motion vector MV_i for the i^{th} VOP relative to the $(i-1)^{th}$ VOP may accordingly be described as given below.

TH-1870_02610203

Algorithm 3.1: Estimation of Motion vector

Given $(i-1)^{th}$ VOP O and the i^{th} VOP I, a set of translated vectors **T** and integers p and j

begin

for $t' = (t_x, t_y) \in$ **T**

Calculate distance transform of edge images O and I.

Calculate $h_{p,t}(O, I)$.

Calculate $h_{j,t}(I, O)$.

Determine $H_t(O, I) = max\{h_p(O, I), h_j(I, O)\}$.

end.

Find $min\{H_t(O, I)\}$ over $\vec{t} \in$ **T**.

Note the translation vector $t' = (t'_x, t'_y)$ corresponding to $min\{H(O, I)\}$.

$MV_i = (t'_x, t'_y)$.

return MV_i

end

3.2.2 Determination of the hand image centroid

In order to form the trajectory from the positions of the hand across the gesture sequence, it is necessary to define a single point on the hand and the trajectory can be obtained by joining the coordinate of this reference point in every frame in the sequence. For this, we choose to use the centroid of the hand as the reference point. This is because even if there is some local motion in the fingers, the centroid will be approximately at the center of the palm and hence the coordinate of this centroid in a frame will best represent the position of the hand in that frame.

TH-1870_02610203

The centroid of the hand in a frame can be easily computed from the first moment of the binary image of the hand. The VOPs obtained are binarized to form binary alpha planes by assigning "**1**" to the hand-pixels in a VOP and "**0**" to the background pixels. After determining the binary alpha plane corresponding to each VOP, the centroid of the hand in each VOP in the gesture sequence is determined using 0^{th} and 1^{st} order moments. The 0^{th} and the 1^{st} moments are defined as [Jain (1989)]:

$$M_{00} = \sum_x \sum_y I(x, y), \quad M_{10} = \sum_x \sum_y xI(x, y), \quad M_{01} = \sum_x \sum_y yI(x, y) \tag{3.1}$$

The centroid $c = (x_c, y_c)$ is calculated as

$$x_c = \frac{M_{10}}{M_{00}} \qquad \text{and} \qquad y_c = \frac{M_{01}}{M_{00}} \tag{3.2}$$

In the above equations, $I(x, y)$ is the pixel value at the position (x, y) of the image. Since the background pixels are assigned "**0**", the centroid of the hand in a frame is essentially the centroid of the binary alpha plane. Therefore, in moment calculation, we may either take the summation over all pixels in the frame or over only the hand pixels.

Once the centroid of the hand in the first VOP is known, the centroids in all subsequent VOPs can also be determined using the motion information obtained above. The centroid in a VOP, say the i^{th} VOP, is determined by shifting the centroid of the previous VOP by the motion vector MV_i corresponding to the current frame. However, slight changes in the shape of the hand in successive frames may cause shifting of actual centroids from that computed from the motion vectors, as shown in Figure 3.3. In view of this, an estimate for the centroid c_i of the i^{th} VOP is obtained by taking the average of the two centroid values computed from the moments and the motion vector respectively[1]. The averaging is done to reduce the variance of the estimation error. This is illustrated in Figure 3.4.

[1]This work has been published in *Springer-Verlag 2005* (Refer item 2 in Page 198 for details)

TH-1870_02610203



Figure 3.3: Centroids calculated by motion vector and moment equations in VOPs.

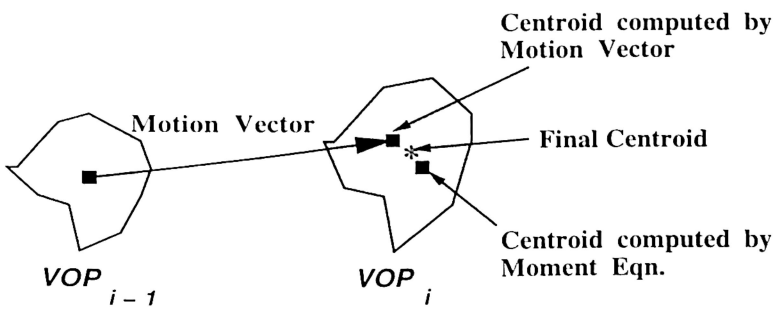


Figure 3.4: Pictorial illustration for estimation of centroids in VOPs.

3.2.3 Trajectory formation and smoothing of the final trajectory

The final gesture trajectory is formed by joining all the calculated centroids in sequential manner while preserving the temporal information. However, the trajectory may be noisy due to the following reasons:

- Centroids may be far away from the actual trajectory due to change in hand shape.
- Hand trembling and unintentional movements during gesturing may cause deviation of the centroids.
- All real gesture trajectories are generally open in practice as it is quite unlikely that in the case of closed gestures a signer will be able to complete a trajectory exactly at the point from where he has started. So, the unclosed end points may result in an open trajectory even if the actual trajectory is a closed one.

TH-1870_02610203

Therefore, in order to take care of this, the final trajectory is smoothened out by removing isolated noisy points with thresholding and replacing each point by the mean of the specified point and its two neighbours. Thus (x_t, y_t) is replaced by

$$(x_t, y_t) = ((x_{t-1} + x_t + x_{t+1})/3, (y_{t-1} + y_t + y_{t+1})/3) \tag{3.3}$$

To tackle the third point, the distance between two end points is computed. If it lies within a threshold, the two end points are merged.

3.2.4 Key VOP based trajectory estimation

The computational burden in estimating the hand trajectory may be significantly reduced by selecting the key video frames in the input gesture video sequence. Since during gesturing the human hand generally does not move very fast, the position of the hand in space does not change much from one frame to the next. Moreover, if we consider trajectory points from all the frames, the estimation error in any frame will change the trajectory. The trajectory will follow the estimation error and may not be smooth as a result. So the lines, curves *etc.*, may not be smooth and a somewhat jerky trajectory may be obtained. For example, a square may have sides as if they are serrated. Instead, if we consider trajectory points only from some key frames and subsequently interpolate the curve/line in between, the resulting trajectory will have a smoother appearance.

Therefore, for trajectory formation it may not be necessary to consider all the frames in a gesture sequence. Accordingly, we propose to select few key VOPs determined for the sequence. If the hand moves by a very small amount (both locally and globally) from one frame to the next, then the corresponding VOPs will be more or less alike. Based on this, the key VOPs in a gesture sequence are identified using the Hausdorff distance measure. Starting with the first VOP as the first key VOP, for every key VOP, the next key VOP is selected as the one for which this measure exceeds a predefined threshold. After getting all the key VOPs in a sequence, hand trajectories are obtained by following the same procedure as above, but considering the key VOPs only.

TH-1870_02610203

3.2.5 Template trajectory generation

For trajectory guided dynamic hand gesture recognition, it is necessary to generate a prototype trajectory defining each gesture in the vocabulary to which an input gesture trajectory is to be compared for recognition purpose. For this, a sufficient number of training samples corresponding to a particular gesture are collected from a number of different signers, each signer making the same gesture at different times and under different conditions. The trajectories obtained from all the training sequences are aligned using the DTW technique, as described later in Section 3.4, and the mean trajectory is derived. This forms the prototype template trajectory corresponding to the gesture sign under consideration.

3.2.6 Key trajectory point selection on the template trajectory

All the template trajectories obtained above, one trajectory per gesture in the vocabulary, form the gesture database to which an input test gesture is to be matched during recognition. Each trajectory is stored in the form of an ordered set of coordinates $\{(x_1, y_1), (x_2, y_2), ...)\}$, where (x_i, y_i) is the spatial position of the hand centroid in the i^{th} VOP (key VOP). That means, the number of points in the set equals to the number of VOPs (key VOPs). However, it may not be always necessary to store all the trajectory points. For example, a square gesture trajectory can be conveniently described by just the four corner points. In view of this, we propose to extract few key trajectory points to be stored in the database. This greatly reduces the memory requirement as well as speeds up the trajectory matching process during recognition.

Each key point on the trajectory is defined by its coordinate (x_i, y_i) in 2D space and the corresponding time instant t_i. These key points represent the prominent locations of the hand in the gesture trajectory. The number of key points is chosen by the user in such a way that the global precision and compactness required for the application is met. Variable time intervals between the key points are chosen to match the local trajectory's smoothness. The basic principle behind key trajectory point selection is merging of adjacent approximation intervals in the estimated template trajectory, until the interpolation error exceeds a predefined threshold, as used in MPEG-7 motion trajectory approximation [Manjunath *et al.*

TH-1870_02610203

(2002)]. We applied piecewise approximation of spatial positions of the centroids in successive VOPs following the motion trajectory approximation and representation technique as described below.

Trajectory approximation

- First order approximation:

$$x(t) = x_i + v_i(t - t_i), \text{where } v_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \tag{3.4}$$

- Second order approximation:

$$x(t) = x_i + v_i(t - t_i) + \frac{1}{2}a_i(t - t_i)^2 \tag{3.5}$$

$$\text{where} \quad v_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} - \frac{1}{2}a_i(t_{i+1} - t_i) \tag{3.6}$$

The value of $y(t)$ may also be approximated in a similar manner. Here v_i and a_i represent hand velocity and acceleration respectively and are considered to be constant over the time interval $[t_i,\ t_{i+1}]$. The coordinate pairs (x_i, y_i) and (x_{i+1}, y_{i+1}) are the hand positions at times t_i and t_{i+1}.

The details of MPEG-7 based motion trajectory representation is discussed in Appendix B. We applied the sequential algorithm [Manjunath *et al.* (2002)] for selection of key trajectory points. The algorithm starts with an approximation interval, which is the union of the first two intervals. It sequentially adds the following points to expand this interval, until the interpolation error exceeds a given threshold. The process repeats for every three successive points in the trajectory path. The end points of each expanded interval constitute the key trajectory points.

3.3 Feature Extraction from Template Trajectories

Trajectory guided gesture recognition requires matching/comparing of an input gesture trajectory to each of the prototype gesture trajectories contained in the database. This may be accomplished either by direct trajectory shape matching and/or by trajectory feature matching. In our recognition method, we propose to employ both shape matching as well as

TH-1870_02610203

feature matching, as described in Section 3.4. For feature based trajectory matching, we propose to use some static and dynamic features extracted from a given gesture trajectory[1]. Static features relate to the shape of the trajectory while the dynamic features relate to the nature of hand motion during gesturing. The static features considered in our work correspond to the total length of the extracted trajectory, location of key trajectory points and the orientation of hand in the gesture trajectory. The only motion information proposed for trajectory classification is velocity. However, we also consider the acceleration feature which is used not for trajectory matching but for movement epenthesis determination in continuous gestures, as described in Chapter 5. The choice of all these features is based on observations over a large number of gesture samples. We observed the nature of hand motion when gestures are performed by different signers and the above mentioned features are proposed after analyzing their behaviours. We define the static features as low level features while dynamic features are high level features. This is because, even if the static features correctly match during classification, *i.e.*, even if the shape features match, it may not represent actual hand trajectory until motion parameters are compared. Motion features represent the dynamics of hand motion in the gesture trajectory. Therefore, for correct and precise recognition, both low level and high level features are required to be considered during trajectory matching.

3.3.1 Static features

Static features are derived from the contour of the gesture trajectory in space. Therefore, these features are related to the shape of the gesture trajectory pattern. However, the overall size of the gesture trajectory may differ from one signer to another. Also, it is possible that different signers make gestures at different positions in the camera frame. This also applies to gestures performed by the same signer but at different times and under different conditions. Accordingly, for truthful recognition of gestures, the algorithm demands for features that are invariant to size and position, *i.e.*, scale and translation (ST) invariance is desired.

[1] This work has been published in *IEEE CIS – RAM 2006* (Refer item 12 in Page 199 for details)

TH-1870_02610203

Translation invariance can be easily obtained by considering the positions of the trajectory points relative to one reference point defined with respect to the trajectory pattern. The reference point that we have chosen is the center of gravity defined as

$$x = \frac{1}{N+1}\sum_{i=0}^{N} x_i \text{ and } y = \frac{1}{N+1}\sum_{i=0}^{N} y_i \tag{3.7}$$

where (x_i, y_i) is the i^{th} key trajectory point, and $N+1$ is the total number of key points.

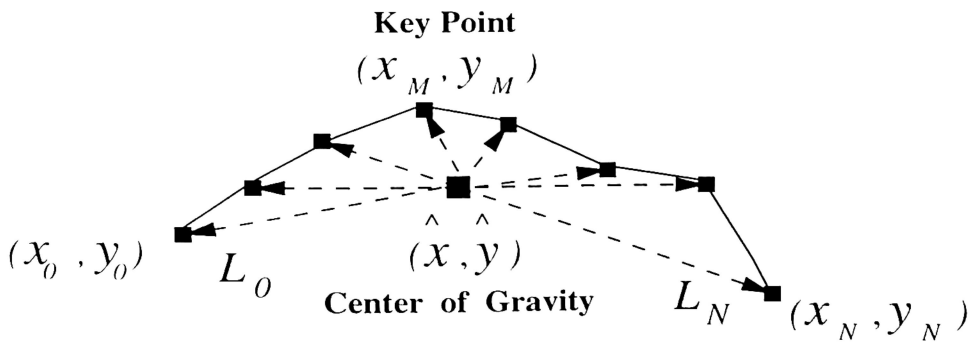


Figure 3.5: Distances of the key trajectory points from the center of gravity.

For scale invariance, it is necessary to calculate the overall size of the gesture pattern in space and then normalize the extracted feature values with respect to the pattern size. This size is given by, as we propose, the average positional distance of all the key trajectory points from the center of gravity, calculated as

$$L_{avg} = \frac{1}{N+1}\sum_{i=0}^{N} L_i \tag{3.8}$$

where the distance of a key point from the center of gravity is simply the Euclidean distance between them, *i.e.*,

$$L_i = \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2} \tag{3.9}$$

This is depicted in Figure 3.5. The static features that we propose to use are –

TH-1870_02610203

- Trajectory length, and
- Number of significant curves on the trajectory.

Trajectory length

For determining the trajectory length (*i.e.*, the total length traversed by the hand during a gesture), the sum of all the Euclidean distances between all pairs of consecutive key trajectory points is calculated.

$$l = \sum_{i=0}^{N} \{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2\}^{\frac{1}{2}} \tag{3.10}$$

where $N + 1$ is the total number of key points on the trajectory. However, for a closed trajectory, the starting point (x_0, y_0) is same as the end point (x_N, y_N) and so only N number of distinct key trajectory points are available.

The trajectory length calculated above will vary with the size of the gesture pattern. So, in order to make it size-invariant, the size-normalized trajectory length is computed by dividing the calculated absolute trajectory length l by L_{avg} obtained in equation (3.8).

For a well-defined gesture pattern, this normalized trajectory length is generally fixed. For example, the normalized trajectory lengths in case of ideal gesture patterns representing "One", "Square" and "Circle" are 2, $4\sqrt{2}$ and 2π, respectively. However, real gestures generally do not follow the ideal trajectories but it is expected that this feature value will be near about the ideal one.

Number of significant curves on the trajectory

This feature is given by the number of key points on the gesture trajectory at which the curvature of the trajectory exceeds some predefined value giving rise to some sharp curves in the trajectory. That means, these are the points at which the hand changes the direction of motion by a significant amount. In order to extract this information, the direction of hand motion at every point on the trajectory is calculated as

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right), \quad i = 1, 2, \ldots, N \tag{3.11}$$

TH-1870_02610203

The number of significant curves (N_θ) in the gesture trajectory is then obtained as the number of points at which the change in direction of hand movement exceeds some predefined threshold T_θ, i.e., $|\theta_{i+1} - \theta_i| \geq T_\theta$. From a large number of visualization experiments, it is observed that a human can generally perceive change in direction of hand movement only when the amount of angular displacement is approximately 45^o or more. Below this, confusion occurs in recognition. Accordingly, we select the threshold T_θ equal to 45^o.

3.3.2 Dynamic features

Motion features are computed from the spatial positions of the hand in the gesture trajectory and the time interval between two prominent hand positions. The two motion features, *viz.*, the velocity (v_i) and the acceleration (a_i) of the hand during gesturing at different positions in the trajectory are computed as given below.

$$v_i = \left\{ \frac{x_{i+1} - x_i}{t_{i+1} - t_i}, \frac{y_{i+1} - y_i}{t_{i+1} - t_i} \right\}, i = 0, 1, 2, ...N - 1 \tag{3.12}$$

$$a_i = \frac{v_{i+1} - v_i}{t_{i+1} - t_i}, i = 0, 1, 2, ...N - 2 \tag{3.13}$$

In most situations, the velocity feature plays a decisive role during the gesture recognition phase. It is based on some of the important observations as listed below.

1. The magnitude of the velocity (speed) is generally more or less constant over a smooth trajectory, *e.g.*, gestures representing "One", "Circle", etc.

   On the other hand, large variation in speed is observed in trajectories with sharp changes in direction, *e.g.*, gestures representing "Square", "Wavy hand", etc.

2. Gestures with linear trajectories (straight line segments) are generally performed at approximately constant high speed, *e.g.*, the gesture representing "One".

   On the other hand, gestures with smooth non-linear trajectories (arcs, circles, etc.) are generally performed at approximately constant but medium speed, *e.g.*, gesture representing "Circle".

TH-1870_02610203

3. In gestures with sharp changes in trajectory direction, the speed of the hand generally becomes very low at points of such sharp changes. Ideally, the hand pauses (speed becomes zero) at these points.

   Accordingly, we have the following deductions:

   - The average speed is generally medium or low – the more the number of direction changes, the lower is the average speed.
   - The mean deviation of the speed from the average speed is generally large.
   - The speed of the hand becomes very low at points of sharp changes in the trajectory direction.

Based on the above deductions, we propose the following features for defining the dynamics of a gesture trajectory.

1. Average speed v_{avg} over the whole trajectory length.
2. Standard deviation σ_v of the speed over the whole trajectory length.
3. Number of minima $N_{v_{min}}$ in the velocity profile that have speed below some threshold $T_{v_{min}}$.

Note that for any gesture the hand always starts from a pause and ends in a pause state. To take care of this, the starting and ending points in the velocity profile are not considered in determining $N_{v_{min}}$. Also, the threshold $T_{v_{min}}$ is generally chosen very small so that for smooth trajectories the number $N_{v_{min}}$ is zero.

The above three features give the characteristics of the hand motion while forming different gesture patterns in space. For example, the trajectory for "One" gesture will give high v_{avg}, low σ_v and $N_{v_{min}} = 0$. Similarly, "Circle" will give medium v_{avg}, low σ_v and $N_{v_{min}} = 0$. On the other hand, a "Square" gesture will give medium or low v_{avg}, high σ_v and $N_{v_{min}} = 3$. Figure 3.6 shows the typical velocity profiles for gestures representing "Circle" and "Square" respectively.

From a number of experiments, we find that the acceleration feature does not play much significant role for trajectory classification/matching. However, this feature may be useful in discriminating co-articulation phase in a continuous dynamic gesture sequence. During

TH-1870_02610203

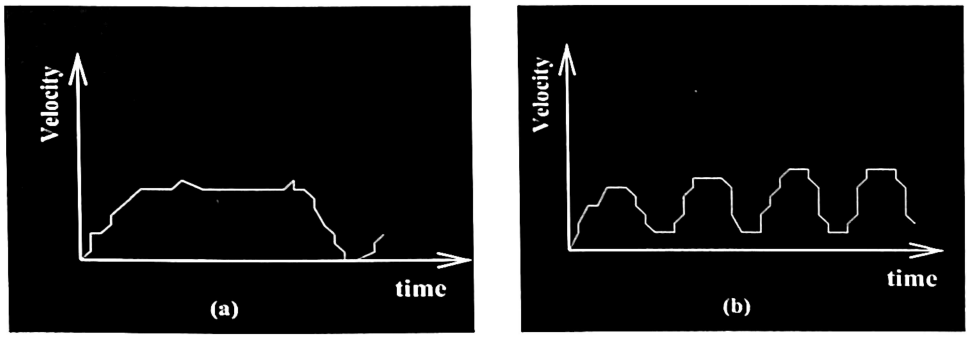


Figure 3.6: Typical velocity plots of gestures representing (a) Circle and (b) Square.

continuous gesturing, one gesture follows another in a sequence. It is observed that during transition from one gesture to the next the hand generally moves with high acceleration from the end point of one gesture to the start position of the next gesture. Based on this observation, we explore the use of the acceleration feature for movement epenthesis detection in Chapter 5.

3.3.3 Forming prototype feature vectors and knowledge-base for gesture matching

The set of feature values extracted from the template trajectory for a particular gesture, as described above, forms the prototype feature vector that gives the mathematical description for that class of gesture. As pointed out earlier, the selected features are trajectory length (l), number of significant curves on the trajectory (N_θ), average speed (v_{avg}), standard deviation of the speed (σ_v) and the number of minima in the velocity profile ($N_{v_{min}}$). Thus, the feature vector is given by

$$\mathbf{F} = \begin{bmatrix} \frac{l}{L_{avg}} \\ N_\theta \\ v_{avg} \\ \sigma_v \\ N_{v_{min}} \end{bmatrix} \tag{3.14}$$

TH-1870_02610203

On the other hand, different features may be weighted according to the degree of their importance. Thus the modified feature vector $\tilde{\mathbf{F}}$ having w_i weight value for i^{th} feature may be written as:

$$\tilde{\mathbf{F}} = \begin{bmatrix} w_1 & \frac{l}{L_{avg}} \\ w_2 & N_\theta \\ w_3 & v_{avg} \\ w_4 & \sigma_v \\ w_5 & N_{v_{min}} \end{bmatrix} \tag{3.15}$$

Note that it is not always mandatory to consider all these features for classification. Depending on the nature of gesture trajectory different feature combinations and weights may be considered.

Finally, all prototype feature vectors, one per gesture class, together form the knowledge-base which is used for gesture matching during classification, as described in the next section.

3.4 Proposed Method for Gesture Classification

3.4.1 Gesture trajectory estimation

For an input test gesture, the first step in recognition involves gesture trajectory estimation. The method employed for the purpose is same as that during training and as described in Section 3.2 above.

3.4.2 Shape based trajectory matching

As a first level of gesture classification, the shape of the gesture trajectory obtained above is compared with each of the template (prototype) trajectories in the knowledge-base. The measure used for trajectory shape matching is maximum boundary deviation (MBD) which is defined as the maximum of the distances of all pixels in the input trajectory to their

TH-1870_02610203

nearest pixel in the template trajectory, expressed as

$$\text{MBD} = \max_{p_i \in P} \min_{q_k \in Q} \| p_i - q_k \| \tag{3.16}$$

where $\| \cdot \|$ represents Euclidean distance, P represents the set of points on the input trajectory, and Q represents the set of points on the template trajectory. This is illustrated in Figure 3.7. Essentially, the MBD measure is the Hausdorff distance measure used for shape comparison.

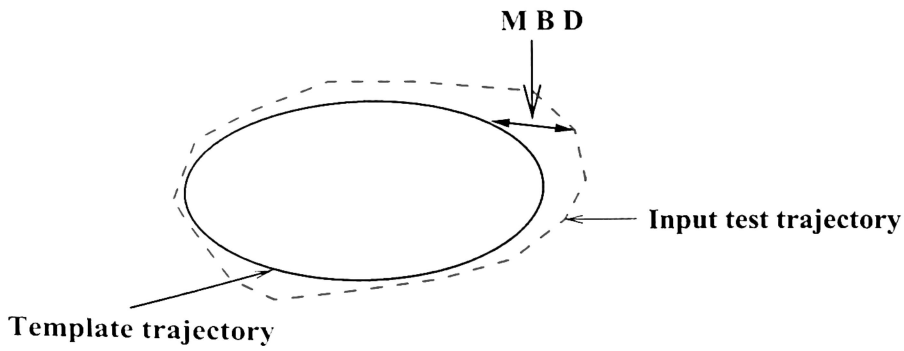


Figure 3.7: Trajectory shape matching.

To allow for certain degree of shape variance in the matched trajectories, a threshold T_s is defined. Template trajectories that have MBD less than T_s with respect to the input trajectory are the candidate trajectories considered for matching in the next stage while all other template trajectories are rejected from further consideration. This reduces the overall computational load.

3.4.3 Nonlinear time alignment and normalization of motion trajectories by dynamic time warping

Dynamic time warping

Dynamic time warping (DTW) is a template-based dynamic matching technique widely used in several algorithms for speech recognition [Rabiner & Juang (1993)]. Even if the

TH-1870_02610203

time scales of a test sequence and a reference sequence are inconsistent, DTW can still successfully establish matching as long as the time ordering constraints hold. Dynamic time warping is a method for computing a nonlinear time normalization between a template vector sequence and a test vector sequence. These two sequences may be of different lengths. For example, Forner-Cordero *et al.* showed that the intrapersonal variations in gait of a single individual can be better captured by DTW rather than by linear warping [Forner-Cordero *et al.* (2006)]. The DTW algorithm, which is based on dynamic programming, computes the best nonlinear time normalization of a test sequence in order to match a template sequence by performing a search over the space of all allowed time normalizations. The space of all time normalizations allowed is judiciously constructed using certain temporal consistency constraints. Following is the list of all the temporal consistency constraints that we have used in the DTW implementation for motion trajectory normalization.

- *End point constraint*: The beginning and the end of each sequence should be rigidly fixed. For example, if the template sequence is of length I and the test sequence is of length J, then only those time normalizations that map the first point of the template to the first point of the test sequence and also map the I^{th} point of the template sequence to the J^{th} point of the test sequence are allowed.
- *Monotonicity constraint*: The warping function that maps the test sequence time to the template sequence time should be monotonically increasing. In other words, the sequence of "events" in both the template and the test sequences should be same.
- *Continuity constraint*: The warping function should be continuous.

Dynamic programming is used to efficiently compute the best warping function and the global warping error. Figure 3.8 demonstrates the matching technique in DTW where correspondence between points in two curves is determined.

Time alignment and normalization by DTW and key point selection

The pattern-matching technique for gesture recognition needs to be able to compare sequences of motion features of the extracted trajectory. The problem associated with the

TH-1870_02610203

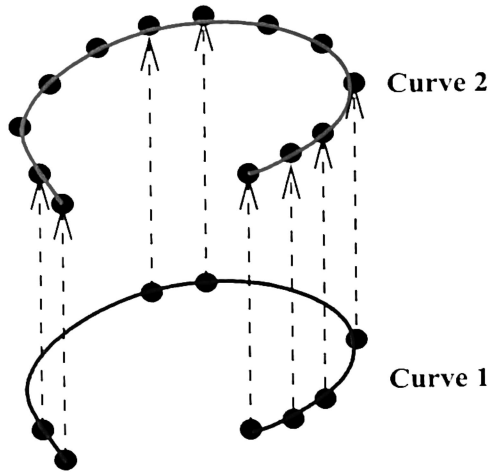


Figure 3.8: Correspondence between points in two curves in DTW.

comparison of motion features of a gesture trajectory arises from the fact that different motion features are seldom realized at the same speed across different instances of the same gesture. Hence, when comparing different tokens of the same gesture, variations due to difference in gesturing speed should not contribute to the dissimilarity score. Variation in speed and time of performing a particular gesture results in variation in length of the extracted trajectory. Thus, there is a need to normalize these fluctuations prior to the comparison of motion trajectory. The fundamental point is that finding the "best" alignment between a pair of patterns is functionally equivalent to finding the "best" path through a grid mapping the motion features of one pattern/trajectory to those of the other pattern. Finding this best path requires solving a minimization problem to evaluate the dissimilarity between the two motion trajectories. This may be accomplished by the DTW algorithm. The goal of the DTW algorithm is to find an optimal time-alignment between two patterns **R** (Reference) and **T** (Test) by evaluating various permitted pairings between the points of the two sequences, and selecting the best alignment path through these points based on some optimality criteria and search constraints [Sakoe & Chiba (1978), Rabiner & Juang (1993)]. A basic version of the DTW algorithm is illustrated in Figure 3.9. In this figure, the two axes represent the trajectory points in sequence, where the horizontal axis corresponds to the input test trajectory **T** and the vertical axis corresponds to the reference trajectory **R**. There are many possible pairings of these points subject to the constraints of monotonicity,

TH-1870_02610203

continuity, boundary alignment, and search window width, as expressed by the following conditions on the arrays i and j of the indices of **T** and **R** respectively.

- Monotonicity condition: $i(k-1) \leq i(k)$ and $j(k-1) \leq j(k)$
- Continuity condition: $i(k) - i(k-1) \leq 1$ and $j(k) - j(k-1) \leq 1$
- Boundary condition: $i(1) = 1$, $j(1) = 1$ and $i(K) = I$, $j(K) = J$, where K is the final index and I and J are the total number of points in **T** and **R** respectively.
- Search window condition: $\mid i(k) - j(k) \mid \leq r$, where r is the maximum permitted search window width.
- Slope constraint condition: The path should not be too steep or too shallow. This prevents very short sequences matching with very long ones.

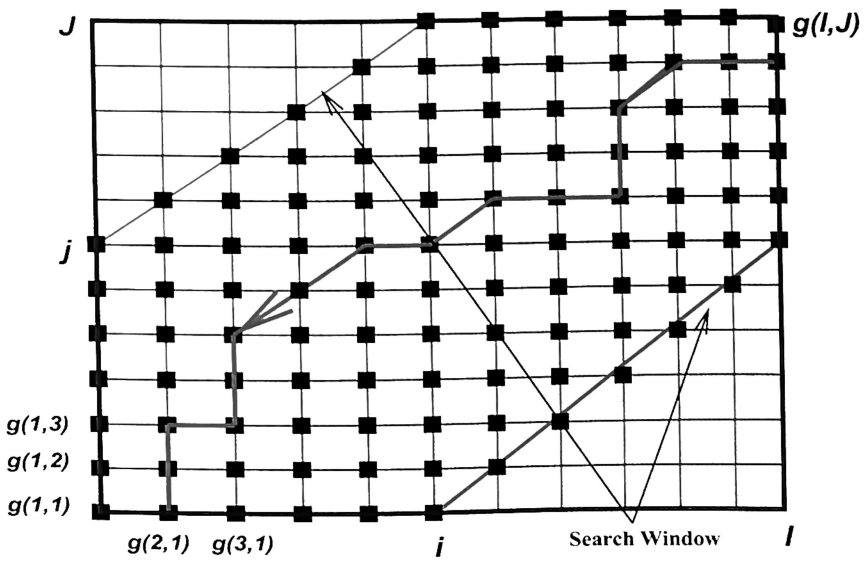


Figure 3.9: Finding the optimal warping function using DTW.

The cumulative distance $g(i,j)$ between the two sequences from the beginning (origin of the grid) of the trajectories to point (i,j) in the grid is calculated as

$$g(i,j) = min \begin{bmatrix} g(i-1,j) + d(i,j) \\ g(i,j-1) + d(i,j) \\ g(i-1,j-1) + d(i,j) \end{bmatrix} \tag{3.17}$$

TH-1870_02610203

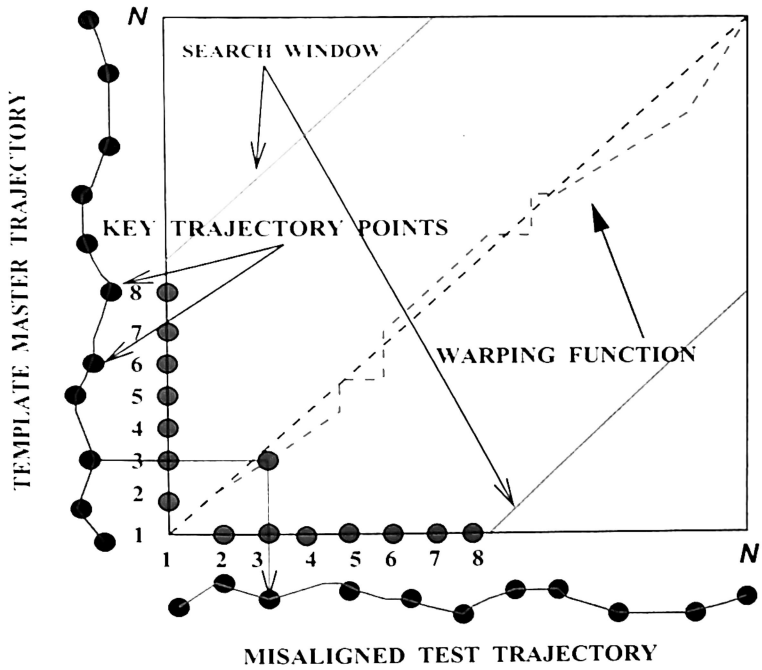


Figure 3.10: Key point selection in the input trajectory by DTW matching.

Here $d(i,j)$ is the distance between the i^{th} point of the first gesture trajectory **R** and j^{th} point of the second trajectory **T** under consideration. The initialization of the above recursion is done by $g(1,1) = d(1,1)$. The overall distance or distortion between the two sequences is given by $D = g(I,J)$, where I and J are the total number of points on the two trajectories. A nonnegative function $w(k)$ is used to weigh $d(i(k),j(k))$ while finding the optimal path. This gives flexibility to the optimal warping path.

The optimal warping function or path is found recursively by starting at point (I,J) and backtracking to the beginning of the gesture trajectories. This is shown by the red coloured line segments in Figure 3.9. The choice of the slope of the path and a weighting function $w(k)$ to weigh $d(i,j)$ in equation (3.17) is to be made for optimal warping [Sakoe & Chiba (1978)]. For our purpose, we select slope equal to zero and a weighting function evaluated as

$$w(k) = \{(i(k) - i(k-1)\} + \{(j(k) - j(k-1)\} \tag{3.18}$$

The basic concept of the proposed trajectory matching under time normalization is shown in Figure 3.10. Here the vertical and horizontal axes represent the spatial axes of a chosen

TH-1870_02610203

template trajectory and that of the input candidate trajectory, respectively. The black dots on the template trajectory indicate the key points chosen for matching. The optimal warping function that establishes the best correspondence between the points of the input trajectory with the master template trajectory is indicated by the red coloured dotted line segments. The key points of the test trajectory are determined by finding points corresponding to the pre-calculated key points of the template trajectory through the optimal warping function[1].

3.4.4 Feature extraction and classification

In the final stage of classification, feature values are extracted from the time aligned (normalized) input gesture trajectory as described in Section 3.3 and the test feature vector is formed as given in equation 3.15.

The feature vector thus formed is compared to each of the prototype feature vectors of the candidate trajectories in the knowledge base except for those gesture trajectories that have been rejected in the first level of classification in the form of trajectory shape matching. The vectors are compared in terms of the Euclidean distance. The input pattern is classified to that gesture class for which the distance measure is minimum. However, if this minimum distance is greater than a threshold T_D then the input gesture is rejected from classification in the sense that the input gesture does not resemble any gesture in the vocabulary.

To estimate the threshold T_D, all the sample training trajectories are treated as the input trajectories. The minimum of the distances of the pairs of feature vectors for each gesture class is determined. Assuming that the training set does not contain any noisy gesture pattern, the maximum of all these minimum distances serves as a good measure of the threshold.

The minimum distance classifier may be alternatively implemented through the Mahalanobis distance (Appendix B). It is to be noted that calculation of Mahalanobis distance requires to know the covariance matrix describing the spread of the feature vectors for each gesture class. This may be estimated during the training phase demanding additional computation.

[1]This work has been published in *IICAI 2005* (Refer item 7 in Page 199 for details)

TH-1870_02610203

The flowchart showing the proposed training and classification algorithms for trajectory guided gesture recognition are shown in Figure 3.11 and Figure 3.12.

3.5 Experimental Results

In our experiment, we considered planar hand gestures generated in front of the camera. The gesture recognition system was implemented on a personal computer with an image capture board. The input images were captured by a CCD camera at a resolution of 120×160 pixels. The gesture videos were recorded at a normal frame rate (30 frames per second). A vocabulary of ten different gestures having only global motion was used. For their simplicity, these gestures may be suitable to several special applications like robot control and gesture based window menu activation in an HCI platform. They are the gestures showing "One", "Two", "Three", "Five", "Six", "Seven", "Alpha", "Square", "Circle" and "Diamond", as illustrated in Figure 3.13. All these gestures are single handed gestures. Each of them are free of *intermediate movement epenthesis*. The proposed algorithm was applied to a database containing 1500 gestures of these classes performed by 10 persons. Each person was asked to do his/her gestures 15 times for each of the 10 gestures. The gesture data were partitioned into two sets, one for training and the other for testing of the proposed algorithm. The partitioning of data is done as shown in Table 3.1. The training and testing gesture data used for our experiment were naturally generated without any constraints on the speed or size of a gesture.

The extracted trajectories for six of these gestures, *viz.*, "Circle", "Square", "One", "Two", "Seven" and "Six" are shown in Figure 3.14. The first row shows the extracted trajectories and second row shows the trajectories after smoothing. Figure 3.15 shows the trajectories obtained using only the key VOPs. Visual analysis shows that the proposed trajectory estimator extract different gesture trajectories as desired for the present purpose. The smoothing operation removes some of the jerks in the gesture trajectories present before smoothing. Further, the gesture trajectories extracted from the key VOPs shows comparatively smoother trajectories. For example, the gesture trajectories "Two", "Circle", "Square" and "Six" extracted from key VOPs appear less noisy than those estimated from

TH-1870_02610203

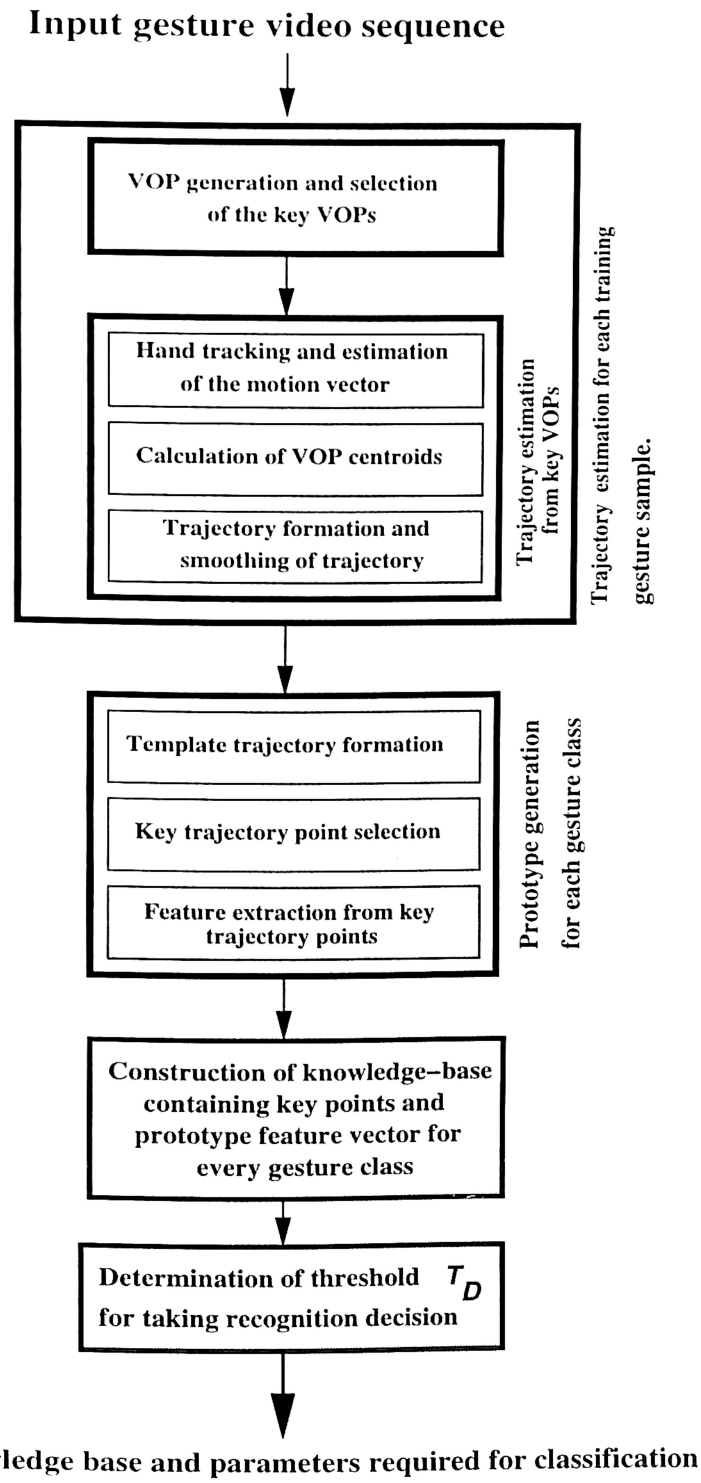


Figure 3.11: Training algorithm.

TH-1870_02610203

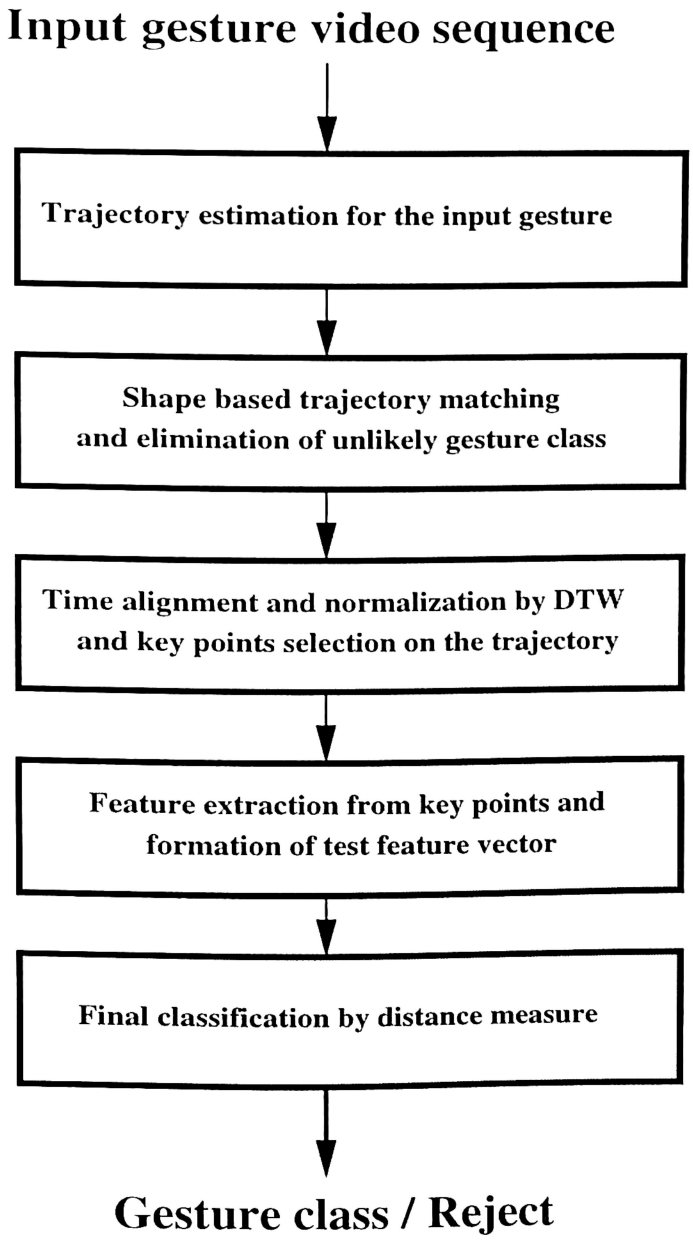


Figure 3.12: Classification algorithm.

TH-1870_02610203

all the VOPs. These motion trajectories encode dynamic characteristics of hand gestures.

As discussed earlier, the shape based trajectory matching selects those trajectories which are similar in shapes with the test trajectory. The feature matching step matches features of the test trajectory to those of the trajectories selected during shape matching. In our experiment, we assigned equal weights of 1.0 to all the component of feature vector $\tilde{\mathbf{F}}$. Table 3.2 shows the classification accuracy in case of the selected gesture vocabulary. In all these cases the proposed recognizer gives recognition accuracy of 90.0% or more with an overall average accuracy of 95.80%, which may be considered as a good recognition rate for HCI applications. For the gesture trajectory "one", v_{avg} is high and $N_{v_{min}} = 0$ and on this basis we achieved cent percent recognition accuracy for this gesture. It is seen that the proposed system gives reasonably good recognition accuracy for gesture trajectories having sharp corners, where the velocity of the hand drops to a minimum value. In all these cases, velocity based features shows good discriminative power in classification. On the other hand, for trajectories with minimum number of corner points, the dynamic features are less dominant. The low recognition rate of gesture trajectories "Circle" and "Six" may be attributed to this. Moreover, it is seen during experimentation that most errors come from the failure of hand extraction that distorts the hand trajectory data. This is because some images were blurry due to rapid hand movements, thus affecting the segmentation of the hand. So, the motion trajectories might not have been extracted correctly.

Recognition rate obtained in our experiments is comparable to most other recognition methods available in the literature. For example, a stroke-based composite HMM method has been used by Kim and Chien for recognizing 3D hand trajectory gestures with an average accuracy rate of 96.88% [Kim & Chien (2001)]. But, the method uses cybergloves. As discussed earlier, Yang *et al.* proposed a method for extraction of 2D motion trajectories based on skin colour and subsequent gesture recognition by TDNN. Recognition rate of 96.21% was obtained on test set of gestures [Yang *et al.* (2002)].

Since, not many vision based approaches for trajectory guided recognition have been reported for the class of gestures considered in this work, our method promises to be a significant development in the field of recognition of dynamic hand gestures having only global motion.

TH-1870_02610203

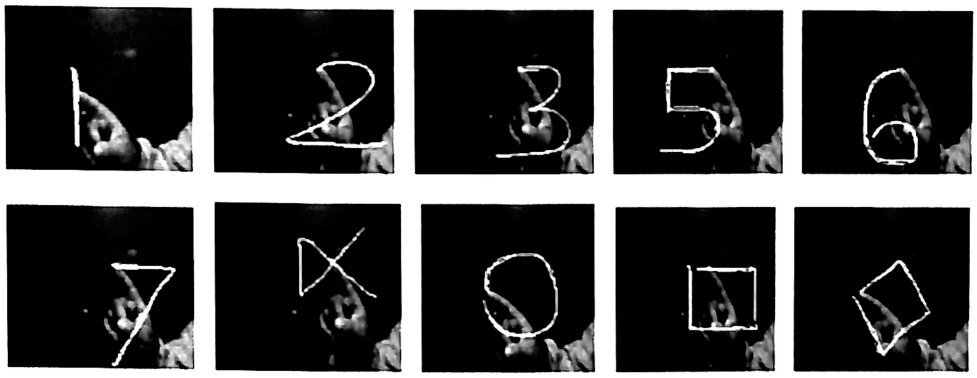


Figure 3.13: Hand gestures showing different motion trajectories corresponding to "One", "Two", "Three", "Five", "Six", "Seven", "Alpha", "Circle", "Square" and "Diamond" respectively.

Table 3.1: Gesture patterns for training and testing.

Gesture	Meaning	Training Data	Test Data
	Circle	60	90
	Square	55	95
	Decision	55	95
	One	50	80
	Two	60	90
	Three	70	100
	Five	60	90
	Six	55	95
	Seven	60	90
	Alpha	60	90
Total		585	915

* indicates start point of a gesture.

TH-1870_02610203

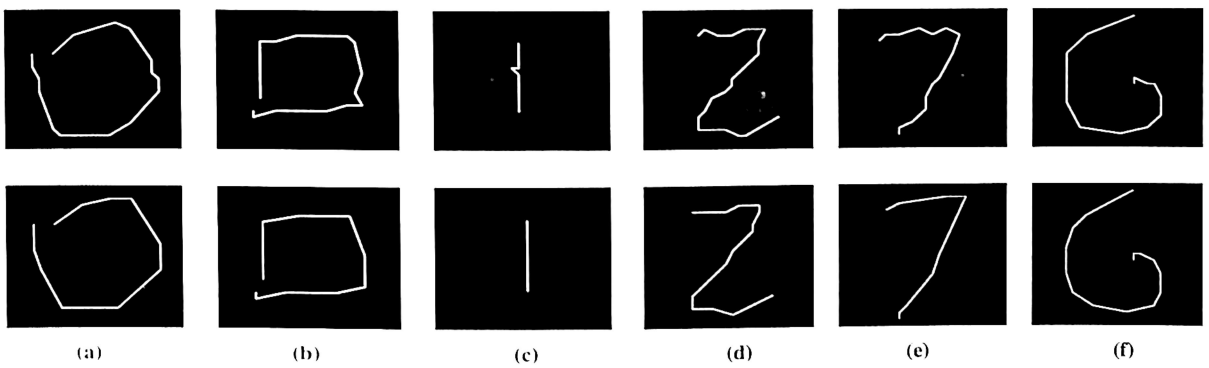


Figure 3.14: Original extracted trajectories are shown in first row and smoothed trajectories for the same are shown in second row of the figure. The gestures represent (a) Circle (b) Square (c) One (d) Two (e) Seven and (f) Six, respectively.

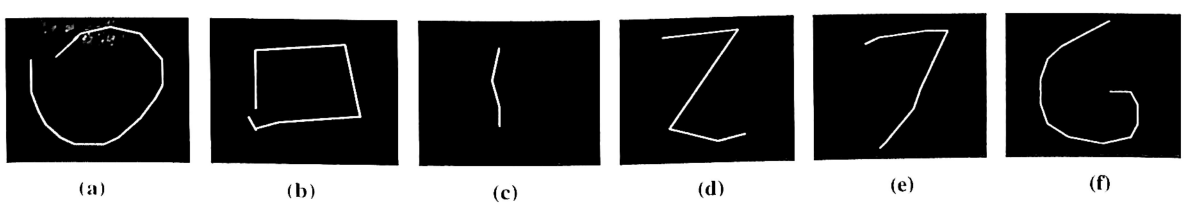


Figure 3.15: Trajectories of (a) Circle (b) Square (c) One (d) Two (e) Seven and (f) Seven, respectively extracted from key VOPs.

TH-1870_02610203

Table 3.2: Experimental results demonstrating the percentage of accuracy in gesture recognition using our proposed method.

Gesture	Meaning	Test Data	Correct	Error	Reject	Recognition (%)
	Circle	90	81	7	2	90.0
	Square	95	92	3	0	96.8
	Decision	95	92	3	0	96.8
	One	80	80	0	0	100.0
	Two	90	88	1	1	97.8
	Three	100	92	2	6	92.0
	Five	90	88	0	2	97.8
	Six	95	87	6	2	91.6
	Seven	90	89	1	0	98.9
	Alpha	90	87	0	3	96.7
Total		**915**	**876**	**23**	**16**	**95.84**

✻ indicates start point of a gesture.

TH-1870_02610203

3.6 Concluding Remarks

A method for trajectory guided dynamic hand gesture recognition is presented in this chapter. The advantage of the system lies in the ease of its use. The user does not need to wear a glove, neither is there the need for a uniform background. Trajectory estimated from the corresponding VOPs in a gesturing sequence bears spatial information regarding hand positions in dynamic gestures and is utilized in the gesture classification stage. Recognition is performed in two stages – unlikely gesture classes are eliminated through simple trajectory point matching in the first stage. This greatly enhances the overall speed of recognition. The second stage of recognition involves feature vector matching. The choice of features in our algorithm is based on observations on the characteristics of real gestures. Experimental results confirm the appropriateness of these proposed trajectory features. However, one limitation of the proposed system is that the DTW has to align the test and prototype trajectories during each classification. This increases the computational load of classification for large vocabulary size.

TH-1870_02610203

Chapter 4

Recognition of Hand Gestures having Local and/or Global Motions

---

The most common form of dynamic hand gestures used in human-computer interaction as well as in sign language communication is gestures having both local and global motions. In general, a gesture may be performed by local hand motion, global hand motion or combination of both. In Chapter 2, we have developed a shape based algorithm for recognition of gestures composed of local motion only. A trajectory matching approach for recognition of gestures formed by global hand motion is presented in Chapter 3. In this chapter, we now propose an algorithm to combine these two recognition schemes in such a way that any gesture can be recognized in a common system irrespective of the nature of hand motion. The proposed system is designed to identify the form of the input hand gesture and subsequently use appropriate algorithm(s) for recognition. The system uses the FSM for representing local hand motion gestures, while the spatial hand position and hand motion features are considered for recognition when the gesture involves global motion. Experimental results show that the proposed system is capable of recognizing a wide class of gestures in a common vision-based recognition system and hence is very much suitable for HCI applications.

TH-1870_02610203

4.1 Introduction

In the previous two chapters, we have developed schemes for recognition of two different kinds of dynamic hand gestures – gestures formed by local hand motions only and gestures formed by global hand motion only. But, commonly used dynamic hand gestures, *e.g.*, hand gestures for sign language, are mostly made up of both local and global motions. Generally the hand makes different shapes while moving along a gesture trajectory. One example of such gestures is shown in Figure 4.1. Gestures of this type are formed by bending one or more fingers and/or turning the palm in different orientations while the hand (arm) as a whole moves in space. The manner in which this shape change occurs provides clues about the finger/palm motions and the activity performed by the hand during gesturing. Recognition of these gestures, hence, should consider the spatio-temporal variations of the hand in the gesture sequence. In simple words, the recognition system should exploit both local and global types of hand motion.

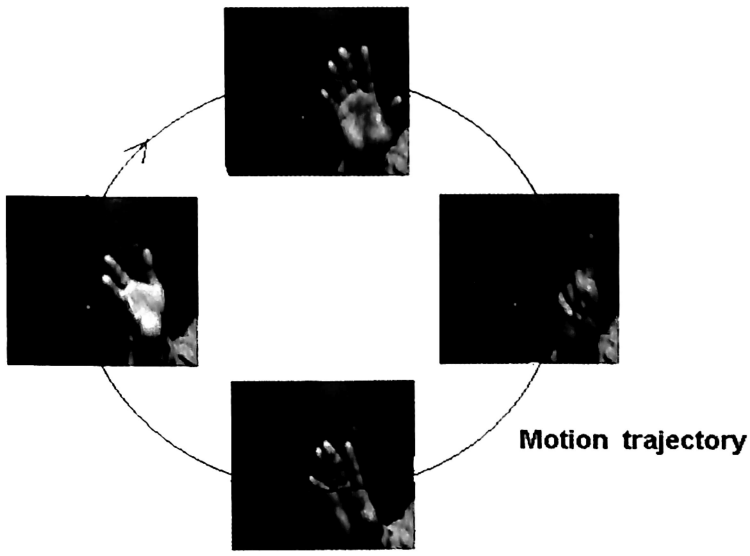


Figure 4.1: Hand moves in a trajectory showing different hand shapes.

The signs used in a sign language present typical examples for gestures having both local and global hand motions. The manner in which shape change occurs provides clues about the finger motions and the activity performed by the fingers during finger spelling.

TH-1870_02610203

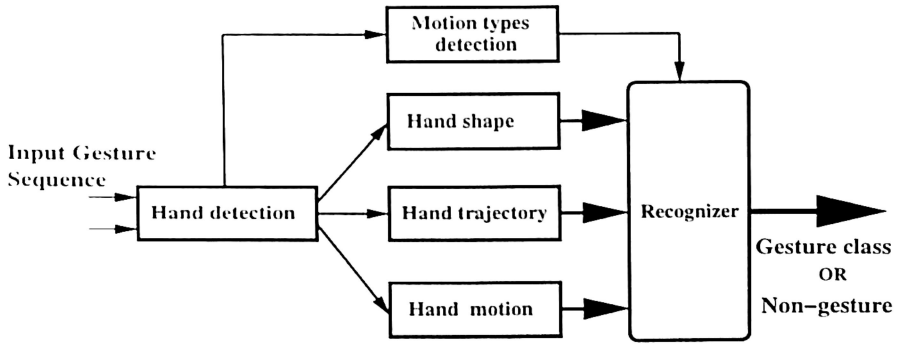


Figure 4.2: Recognition of gestures having both local and global motions.

Sign language recognition by itself is a multidisciplinary research area involving pattern recognition, computer vision, natural language processing and psychology. It is a comprehensive problem because of the complexity involved in the visual analysis of signs and the highly structured nature of sign languages. In addition to hand movement, several components such as the use of facial expressions and head movements, affect the context of a sign language. While the structure of spoken language makes use of words linearly, *i.e.*, one after the other, a sign language makes use of several body movements in parallel in a spatio-temporal space. Thus a recognition system solely based on hand gestures cannot give the full context of the sign language. Most of the sign language recognizers available for different sign languages mainly deal with global hand motion in space while utilizing very limited information on local motion of the hands [Yang *et al.* (2002), Holden *et al.* (2005)]. Because of this, these systems cannot differentiate two signs having same gesture trajectory even if the hand shapes are significantly different.

The gestures normally used in HCI applications contain restricted motions of the hand. This is done with an aim to increase the recognition accuracy by the available recognition techniques [Pavlovic *et al.* (1997)]. But, user friendly HCI applications demand for hand gestures which have both local and global motions. Most of the literature on isolated hand gesture recognition in a vision based setup has not considered the case of gestures having both these components of motion. The proposed integration is motivated by the need for a general recognizer capable of recognizing isolated gestures having different spatio-

TH-1870_02610203

temporal behaviour. It analyzes local as well as global hand motion information to arrive at a recognition decision.

The integrated recognition scheme is outlined in Figure 4.2. It utilizes the hand shape, the gesture trajectory and the hand motion information for recognition. The major step in the proposed integration is the detection of the type of motion of the hand in the gesture sequence. The success of the subsequent recognition system is dependent on the success of this step. We first develop a simple procedure to detect the type of motion. The following sections present the proposed integration.

4.2 Proposed Integrated Gesture Recognition Scheme

Recognition methods developed so far are gesture type specific – the shape based approach is developed specifically for gestures having local motions only while the trajectory guided recognition scheme is applicable in case of gestures having global motion only. However, it is desired to have a recognition system that is capable of recognizing a gesture irrespective of the nature of the hand motion. For this, we develop a scheme which can detect the nature of hand motion involved in the input gesture and then apply recognition algorithm appropriate for that gesture[1].

The proposed scheme incorporates three different basic stages for gesture recognition. They are FSM-based shape matching, trajectory shape matching and matching of gesture trajectory features. As discussed in Chapter 2, local hand motion is quantified by shape information. Thus, a gesture can be modelled in the form of an FSM where each gesture state corresponds to a particular hand pose in the gesture sequence. Consequently, gesture recognition can be accomplished simply by the FSM matching. On the other hand, for gestures having only global hand motions, hand shape information does not play any role in classification. As discussed in Chapter 3, for recognition of this type of gestures, we consider hand spatio-temporal information in the form of a motion trajectory and recognition is performed via trajectory matching. Trajectory matching is done on the basis of trajectory shape comparison followed by matching of some important trajectory features for better

[1]This work has been published in *ICoMMS 2006* (Refer item 14 in Page 199 for details)

TH-1870_02610203

accuracy. Finally, for recognition of gestures having both local and global hand motions, we propose to combine the shape-based and trajectory-based schemes. The basic steps in the proposed recognition scheme are given by the flow-chart in Figure 4.3.

4.2.1 Gesture type detection

The shape of an object can be represented by means of boundary, region, moment and structural description [Jain (1989)]. We can use these representations for making measurements of shapes and decide whether a particular shape has undergone any change from a frame to another frame. The centroid and central moments may be used to conveniently represent many shape features. Our goal being to detect only changes in shape, rather than representing the shape accurately, we propose to use the centroid as the only parameter to detect the change of a hand shape during gesturing.

The spatio-temporal location of the hand in a gesture sequence can be represented by the locations of the centroids in the key VOPs relative to the first key VOP of the sequence. The centroid of a shape represented by a VOP was defined in Chapter 3. The nature of hand motion associated with an input gesture may, therefore, be determined on the basis of displacement of the key VOP centroids in the 2D image plane.

In the case of gestures having local motion only, the motion of the centroids include the following:

- Movement because of the change of shape of the hand; and
- Movement because of hand trembling.
- Movement because of occlusion. This motion of the centroid is not considered in present context.

While the movement of the centroids because of the change of hand shape is limited by the size of the hand, there is a signer-dependent shift because of hand trembling. However, the resulting movement of the centroids is relatively small. On the other hand, in the case of gestures having global motions, the centroids will follow the motion trajectory and as such there will be significant movements of the centroids in the 2D plane. Thus, the spread of the centroids in the 2D plane can be used to identify the type of motion present in a gesture

TH-1870_02610203

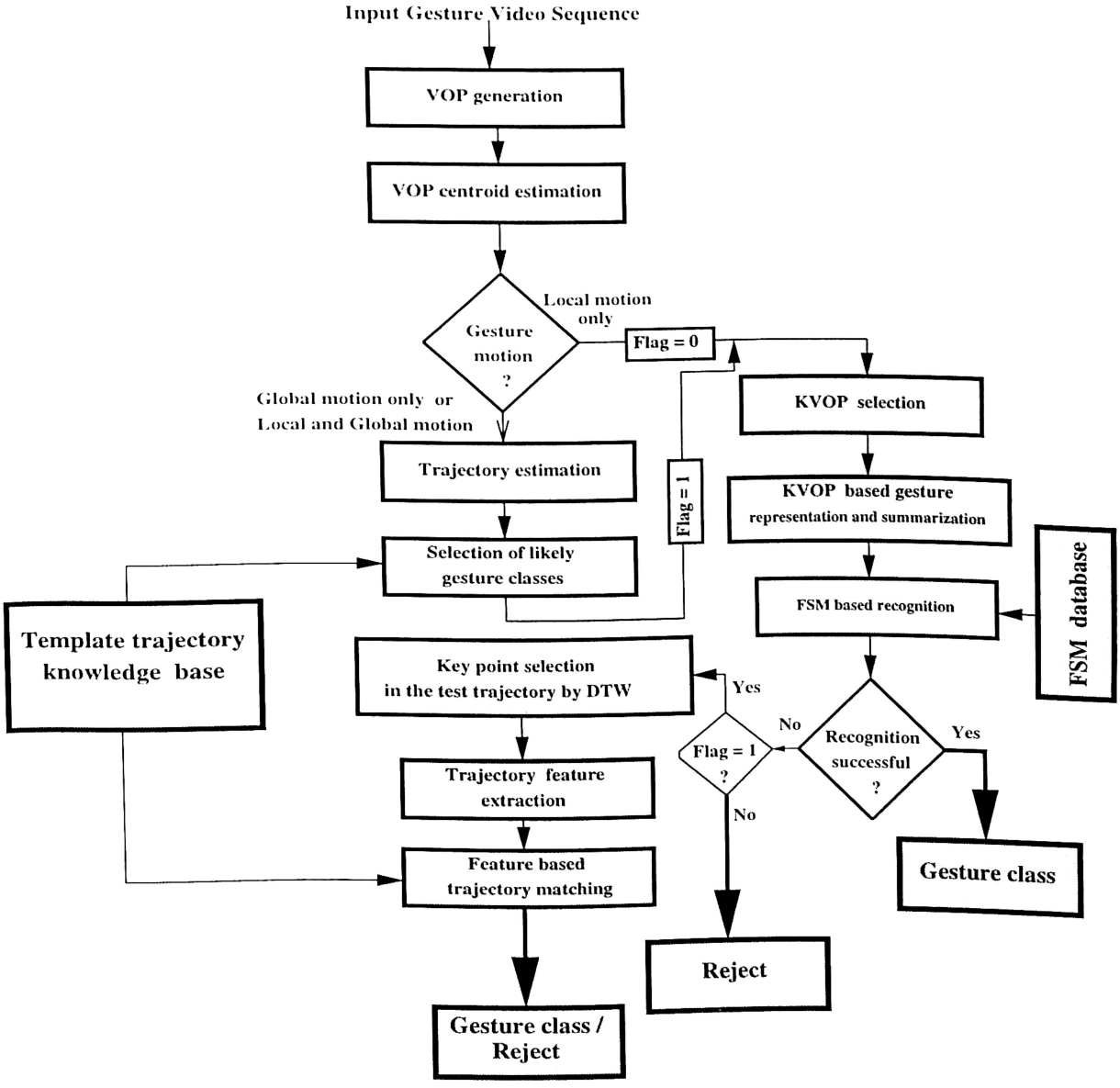


Figure 4.3: Block diagram for the proposed gesture recognition system.

TH-1870_02610203

sequence. Figure 4.4 and Figure 4.5 illustrate the relative movements of the centroids in the case of gestures having local motion and global motion respectively.

The proposed method considers the centroids of the key VOPs rather than those of the VOPs. Because, there is a significant change in the video content from the frame corresponding to one key VOP to the next. A circular region is considered around the centroid of the first key VOP. The diameter of this circular space represents the allowable movement of the VOP centroid on account of hand trembling and hand shape change. If all the VOP centroids are within this circle, as in Figure 4.6, then the input gesture is considered to be composed of local motions only. Else, global movement of the hand is detected. The diameter of the circular region may be determined during the training session.

To consider the abnormal hand trembling/movement and sudden change in the hand shape (which may occurs only in a few key VOPs of the sequence), one criterion may be fixed for taking the decision about the spatio-temporal behaviour of a gesture. If the majority of the centroids lie within the circular region, then the sequence may be considered as a gesture sequence having only local hand motions. Otherwise, the sequence has global hand motions. A statistical measure for the *majority of centroids* can be experimentally determined.

4.2.2 Gesture recognition modules

For gestures having only local hand motions, recognition is performed by the FSM representation of the input gesture sequence, as explained in Chapter 2. The input gesture sequence is summarized on the basis of shape similarity measurement of the video object planes and subsequently represented by a finite number of states along with associated temporal information. Finally, this representation is used for recognition.

For recognition of gestures having only global hand motions or gestures having both local and global hand motions, recognition is accomplished in two or three hierarchical levels as described below.

- **Level 1:** First, the motion trajectory is estimated and smoothened out. After finding the gesture strokes, the next task is to find all those gesture signs in the gesture

TH-1870_02610203

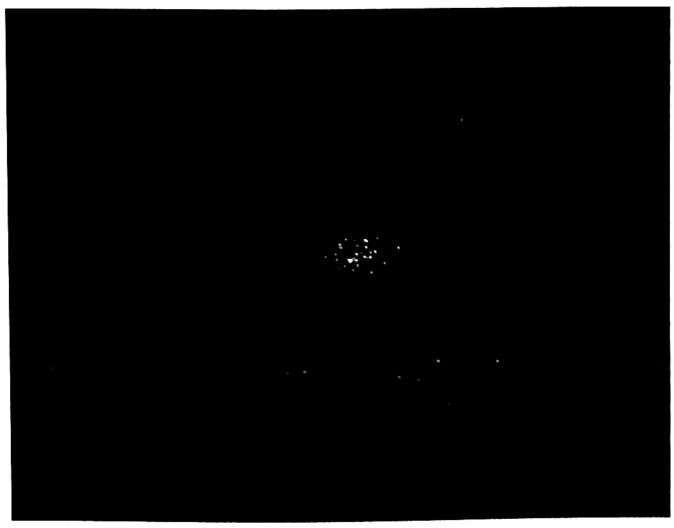


Figure 4.4: Movement of VOP centroids in 2D plane in case of gestures having local motion only.

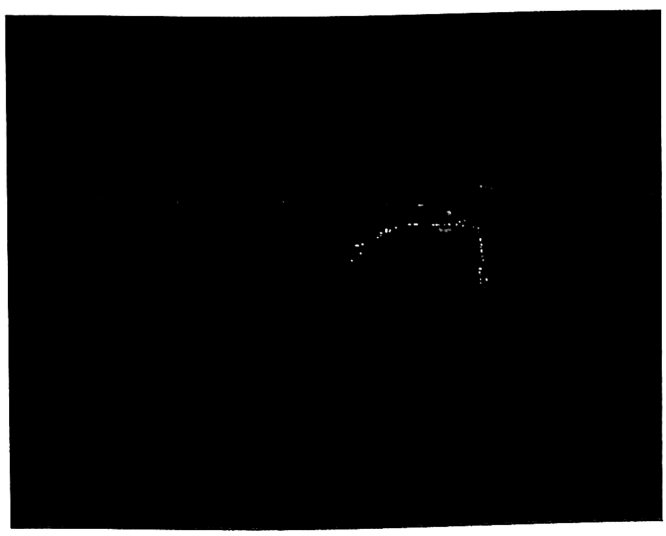


Figure 4.5: Movement of VOP centroids in 2D plane in case of gestures having global motion.

TH-1870_02610203

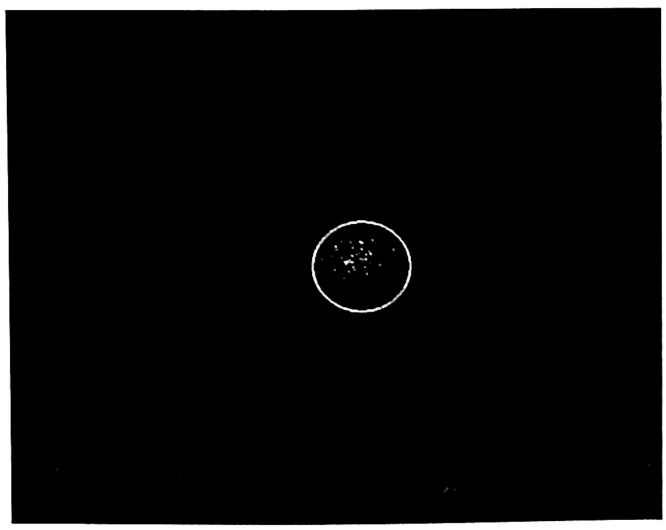


Figure 4.6: Cluster of centroids in 2D plane in case of gestures having local motion only.

vocabulary that have similar trajectory patterns. For this, we measure shape similarity between the input gesture trajectory and the template trajectories contained in our gesture vocabulary database. These template trajectories, one trajectory per sign, are generated and stored during the training phase. This is the first level of recognition hierarchy in which all unlikely gesture patterns whose trajectories are very much different from the input are discarded from further consideration.

- **Level 2:** In the second level, the input gesture is classified on the basis of hand pose and shape (finger movement and palm orientation). This is accomplished via gesture representation by the FSM and subsequent FSM matching for gesture classification, as described in Chapter 2. However, the scheme described in Chapter 2 is for gestures without any global motion whereas in the present case it is required to compensate for the global motion by which the whole hand moves in space. For this, the motion vector to the centroid of each key VOP with respect to the centroid in the first key VOP is determined and the hand is translated back by an amount equal to the motion vector.

It is to be noted that all the gesture patterns shortlisted in the first level are those whose 2D trajectory patterns resemble the trajectory of the input gesture. Some of

TH-1870_02610203

these gestures in the vocabulary may be combination of both global and local motions while the others are composed of global motion only. For the first kind, hand shape variations due to the local motion are modeled in the form of an FSM. In the second stage of recognition, the FSM derived from the input (after compensating for the global motion, as described in the previous paragraph) is matched to the FSM representations available for the second kind of shortlisted gesture patterns. We arrive at a final decision only if all the input states pass successfully during the matching process. Else the input is deemed to be a gesture composed of global motion only and we move on to the third level of recognition.

- **Level 3:** In the third level, as shown in Figure 4.7, the key points on each model trajectory (corresponding to those gestures selected in Stage 1 but with global motion only) are mapped onto the test trajectory by the DTW algorithm. This is to tackle nonlinear time alignment between the test and template trajectories[1]. Thus, the shape similarity measurement and the DTW-based state mapping collectively cope with the problem of spatio-temporal variation. Following this, the static and dynamic features quantifying a given gesture trajectory, as proposed earlier in Chapter 3, are extracted from the input test trajectory pattern. The feature vector thus formed is compared to each of the prototype feature vectors of the candidate trajectories in the knowledge base except for those gesture trajectories that have been rejected in the first level of classification in the form of trajectory shape matching. The comparison is done on the basis of the Euclidean distance measure and the input is classified to that gesture class corresponding to the nearest prototype vector.

4.3 Experimental Details

4.3.1 Experimental data sets

In the first set of experiments, we considered all the ten gestures with only local hand motions as defined in Chapter 2. These experiments validate the proposed method for determination of types of motion present in a gesture sequence.

[1]This work has been published in *NCC 2006* (Refer item 13 in Page 199 for details)

TH-1870_02610203

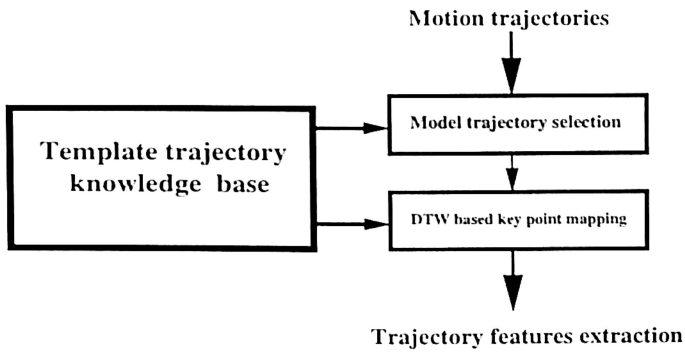


Figure 4.7: Mapping of key points from template to test trajectory.

In the second set of experiments, we selected ten different gestures which have both local and global hand motions. These gestures are shown in Figure 4.8. Here hand moves in space with the hand shape changing continuously while making a gesture. These selected gestures are easy to perform and may be suitable for HCI applications.

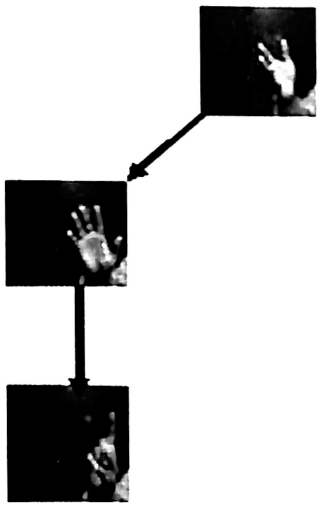
In the third set of experiments, we considered several static and dynamic gesture signs which are selected from the online dictionary of native Indian sign language by William Tomkins. The signs used in the experiments are given in Figure 4.9 and Figure 4.10. Figure 4.10 shows the pictographs of the dynamic signs. Figure 4.11 illustrates five frames of five of the signs used. The set of dynamic gestures considered here includes gestures with either global motion only or both local and global motions.

The experiments with sign language gestures are restricted to hand movements only. Recognition of facial expressions and head movements is beyond the scope of the current research. Furthermore, for the sake of simplicity, we considered dynamic gestures with single hand motion only although it is possible that both hands are involved in generating a particular sign. The proposed recognition technique may be extended to application to such cases.

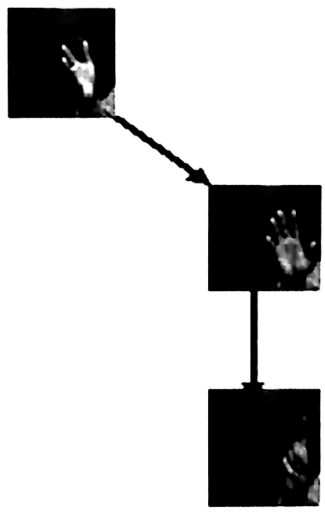
4.3.2 Experimental setup

Figure 4.12 shows the experimental setup and the system command window used in the experiment. The system is implemented on an Intel Pentium IV (HT Technology) based system on Microsoft Windows XP® platform. The hand images are captured by a single

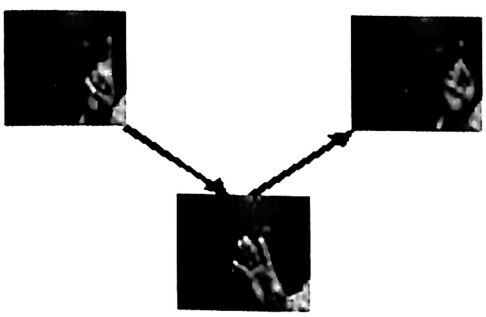
TH-1870_02610203



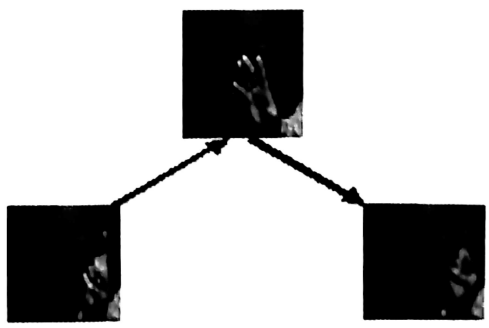
(a) Gesture No. 1



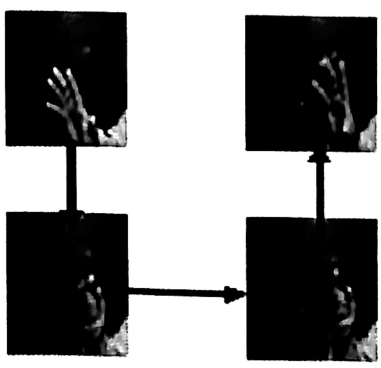
(b) Gesture No. 2



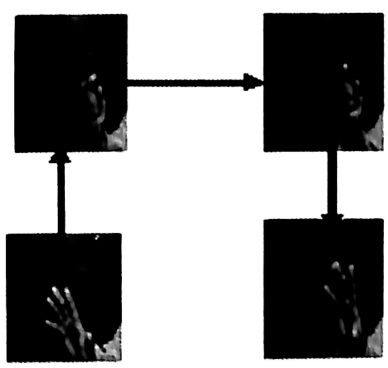
(c) Gesture No. 3



(d) Gesture No. 4

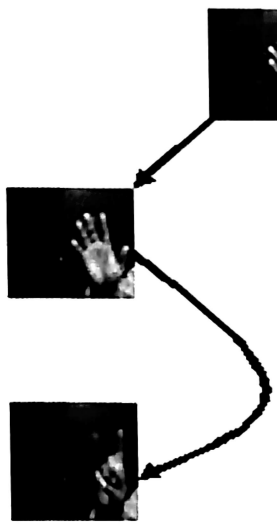


(e) Gesture No. 5

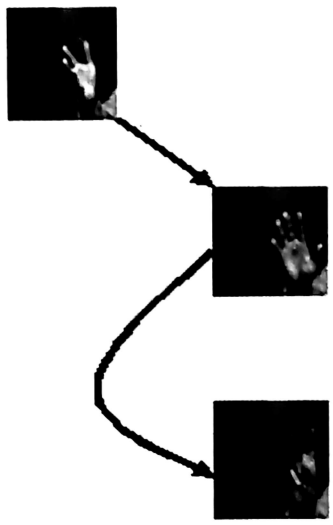


(f) Gesture No. 6

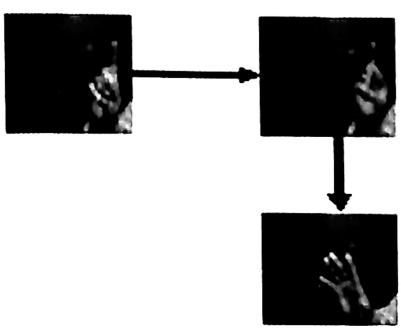
TH-1870_02610203



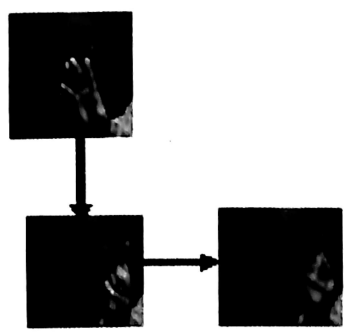
(g) Gesture No. 7



(h) Gesture No. 8



(i) Gesture No. 9



(j) Gesture No. 10

Figure 4.8: Gestures having both local and global hand motions.

TH-1870_02610203

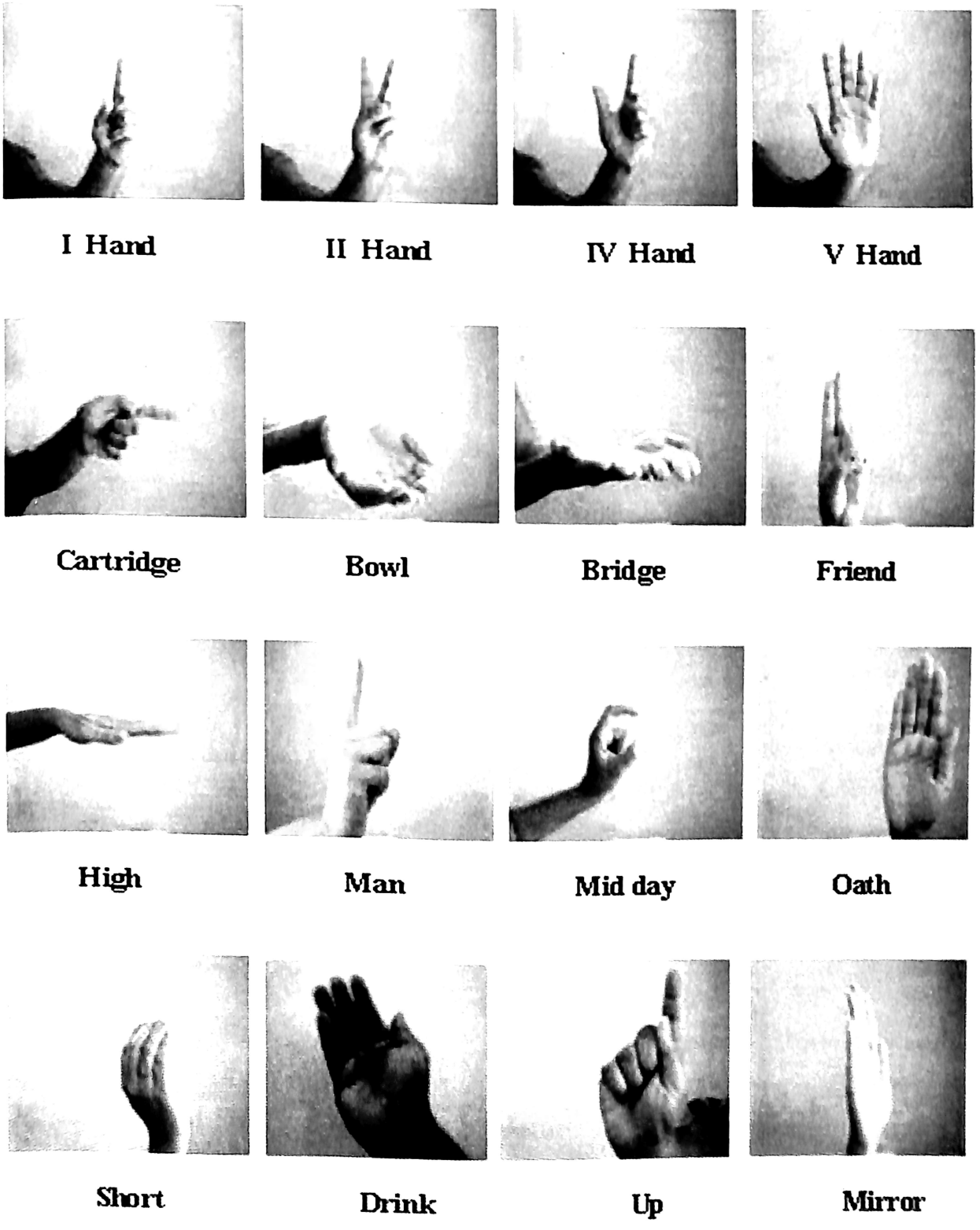
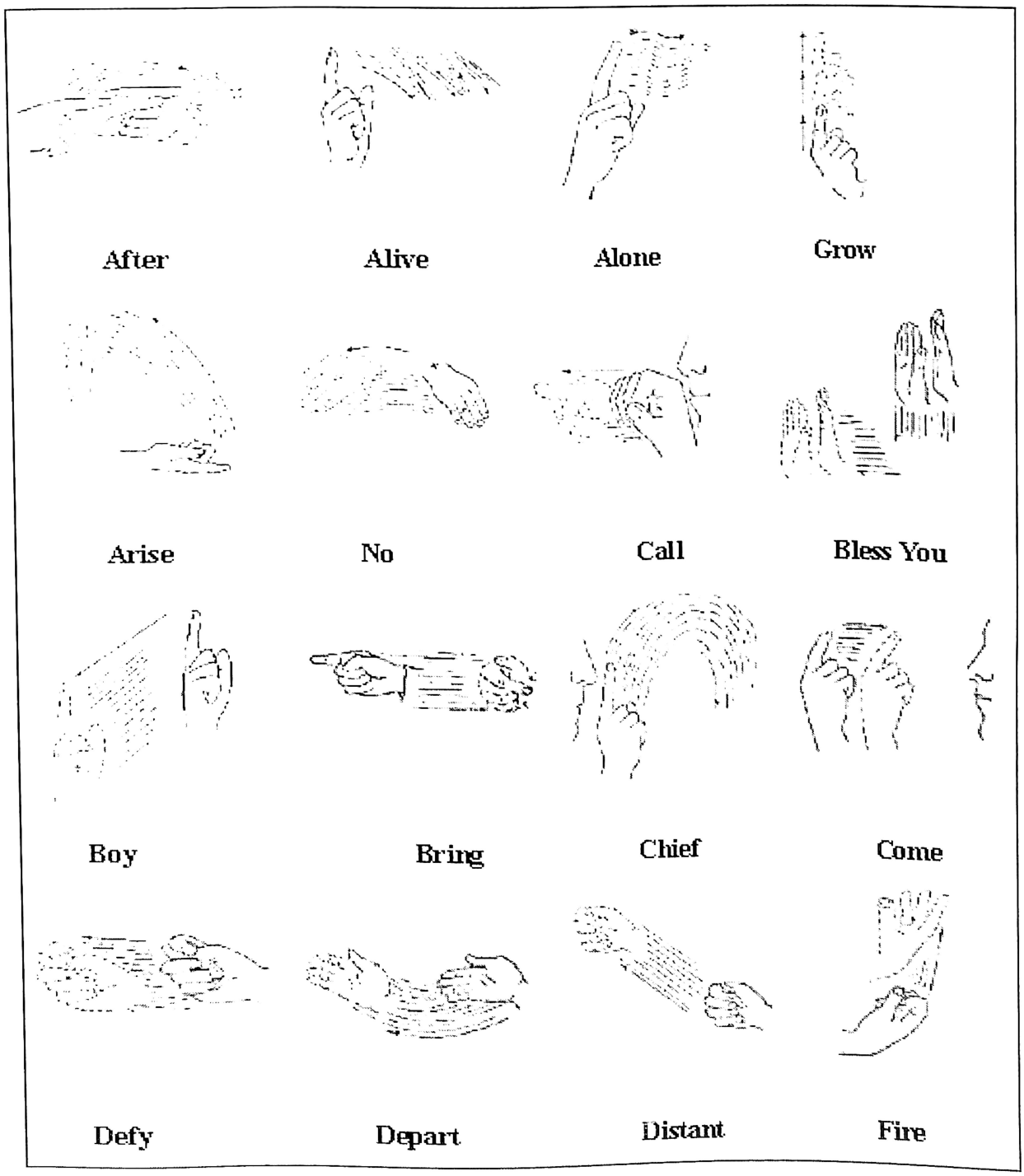


Figure 4.9: Some static hand postures of the native Indian sign language.

TH-1870_02610203



TH-1870_02610203

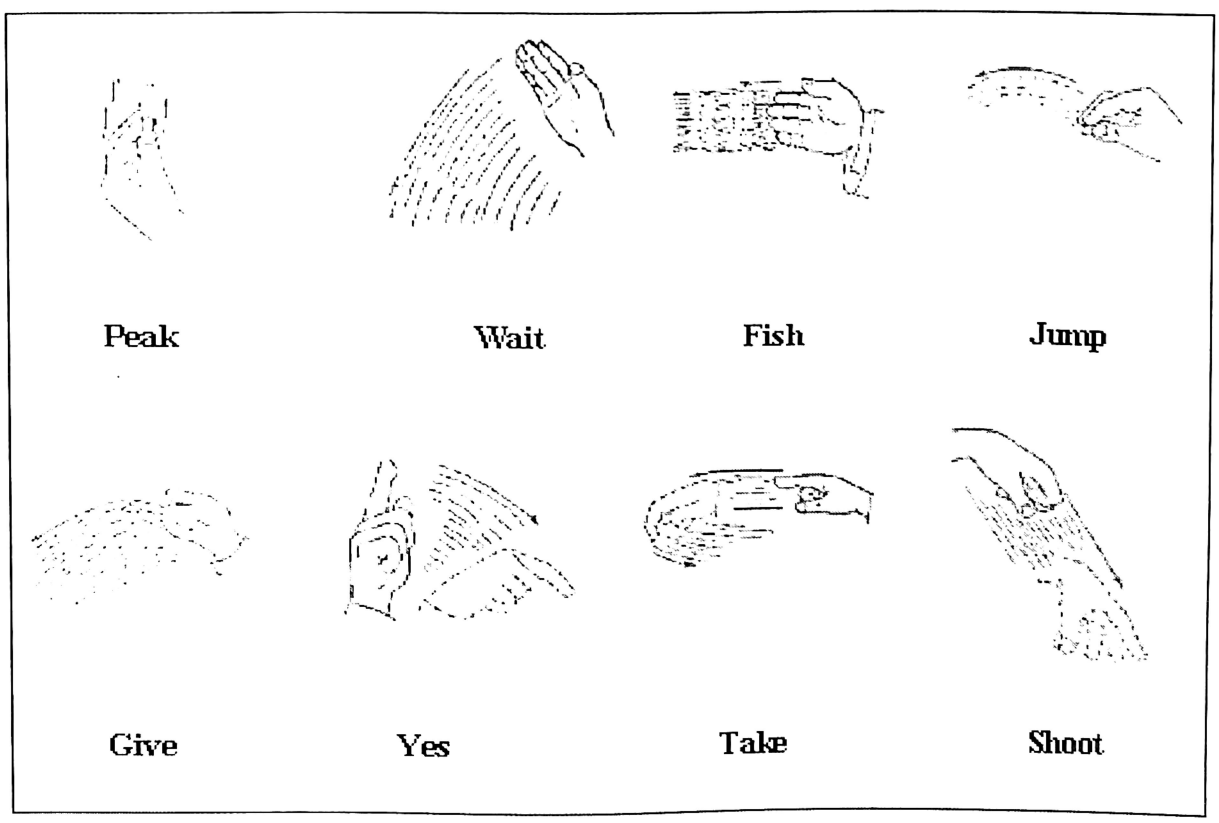


Figure 4.10: Dynamic hand signs (pictographs) of the native Indian sign language taken from William Tomkin's online database.

TH-1870_02610203

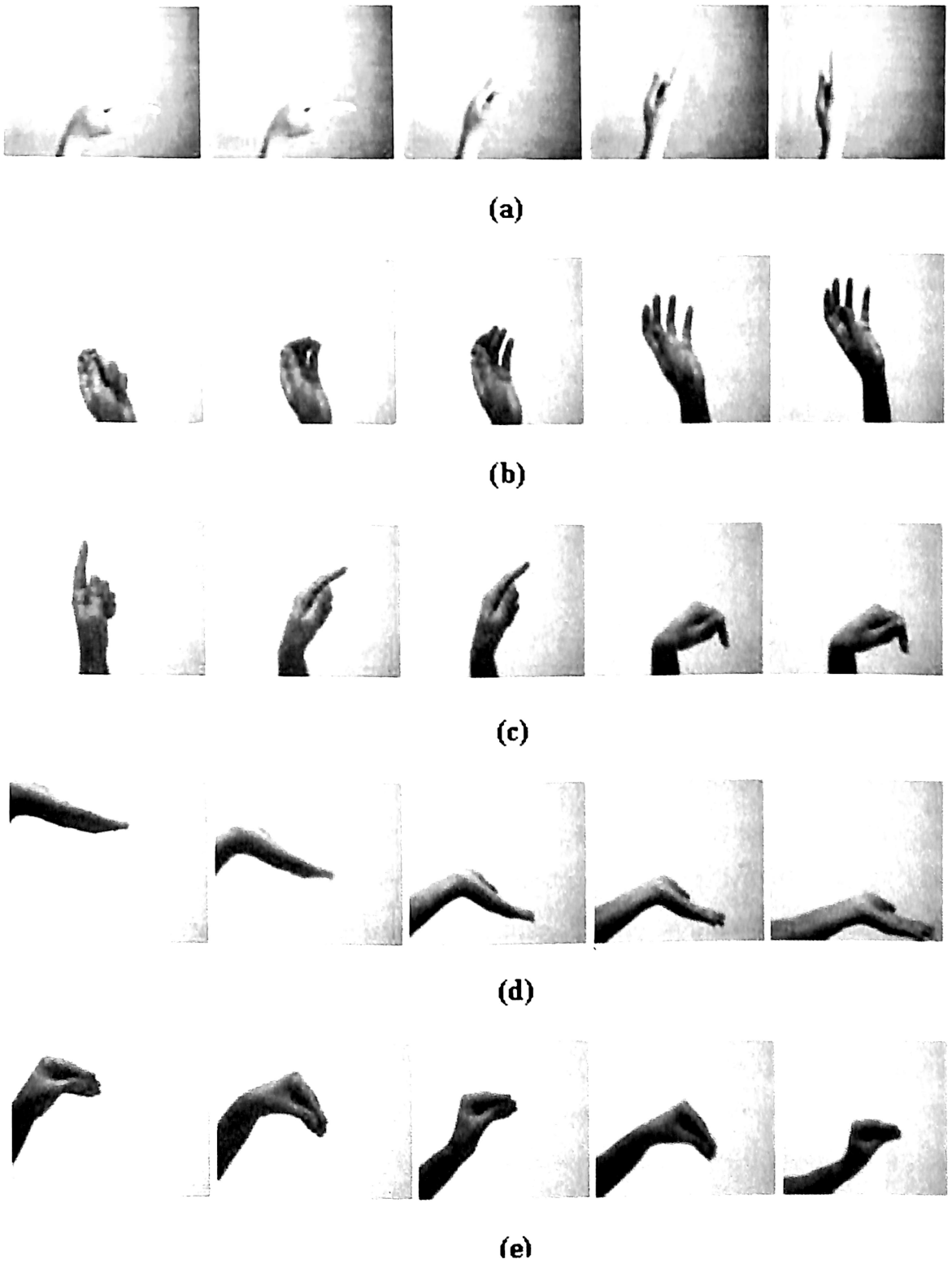


Figure 4.11: Five frames of the signs from the native Indian sign language (a) Arise (b) Fire (c) Yes (d) Give and (e) Jump, respectively.

TH-1870_02610203

camera mounted in such a way that it covers the whole hand action during video acquisition. The videos were acquired as RGB colour video and converted to gray-scale frames for further processing. The system is designed to recognize hand gestures by using both finite state machine and the extracted hand trajectories. As in any recognition system, the system is first trained to adjust all parameters of the recognition system. Recognition is performed by considering hand shapes, spatial hand positions and trajectory based features along with temporal information. Finally, the recognition results are displayed on a command window in the form of a text message indicating the gesture name as identified by the system[1].

4.3.3 Experimental results

Gesture type detection

For the training sequences of gestures considered in Chapter 2, we plotted the centroids of key VOPs in a 2D image plane. We observed a cluster of centroids as shown in Figure 4.13. The cluster indicates that considered gesture sequences have only local motion. A circular region is constructed to cover all the centroids in the 2D image plane. Now, for all the testing gesture sequences considered in Chapter 2, the location of centroids in the 2D plane is observed. It is seen that, for all these testing gesture sequences, centroids of key VOPs lie in the circular region as fixed during the training. For all the gesture sequences (where significant amount of global motions present) considered in Chapter 3, no such cluster formation is observed. In these cases, a significant number of the centroids of key VOPs in the sequence are outside the circular region. Figure 4.14 shows the relative movements of centroids for all the testing samples corresponding to "Square" gesture. It is seen that the proposed centroid based method has a good discriminative power in detecting types of motion present in a gesture sequence. It is to be noted that, the method may fail to judge a global motion in a sequence which has small global motion in all the frames of the sequence. In this case, it is confused with a sequence that has only local hand motion. So, in the present cases we selected gestures that have significant amount of global motions, so that they are no way confused with gestures having only local hand motion.

[1]This work has been published in *IEEE TENCON 2005* (Refer item 9 in Page 199 for details)

TH-1870_02610203



Figure 4.12: Test setup and command window for the proposed gesture recognition system

TH-1870_02610203

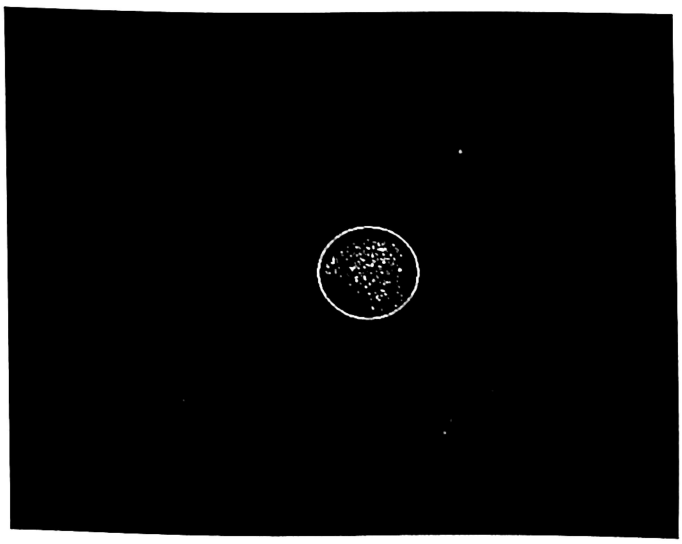


Figure 4.13: Cluster of centroids in 2D plane for all gestures considered in Chapter 2.

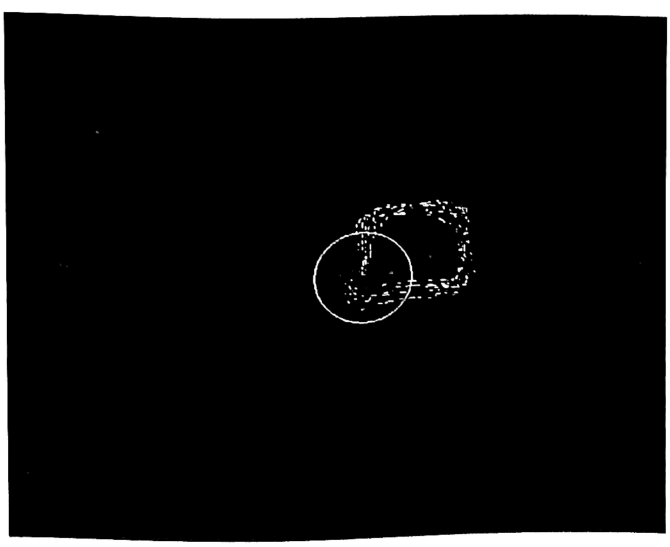


Figure 4.14: Movement of centroids in 2D plane for the "Square" gesture considered in Chapter 3.

TH-1870_02610203

Recognition of gestures having local and global motions

Next step of the proposed recognition system is to pass a gesture sequence to an appropriate recognition engine for classification. Gestures having only local hand motions are classified by the FSM matching, while gestures having only global hand motions are classified by the trajectory guided recognition. Recognition rates for these two cases are given in Chapter 2 and Chapter 3 respectively.

To test the performance of the proposed system for recognition of gestures having both local and global hand motions, the ten different hand gestures as defined in Section 4.3.1 are used. Table 4.1 shows the gesture recognition results for the second set of experiments. The overall recognition rate is 94.3% for the selected gesture vocabulary. The recognition rate so obtained is satisfactory for HCI applications.

Recognition of signs of the native Indian sign language

Table 4.2 shows the recognition rate for the 16 selected hand posture gestures (static gestures) in the native Indian sign language[1]. A static hand pose gesture may be viewed either as a special case of local motion gesture (that is represented by a single-state FSM of infinite duration, as described in Chapter 2), or as a special case of local-plus-global motion gesture (where the gesture trajectory is a single point in space). For these gestures, only appearance and/or shape of the hand is considered for recognition. The overall recognition rate is found to be above 95%. In this case, two handed static gestures ('Bowl' and 'Bridge') show reasonably good recognition accuracy. For recognition of dynamic hand gestures in the native Indian sign language, we selected altogether 24 different signs having both local and global hand motions, each sign representing a single word. Table 4.3 shows the recognition rate for these selected words. The average recognition rate obtained in this case is also above 95%.

Therefore, the gesture recognition module developed in this chapter proves to be very efficient in recognizing a wide class of dynamic hand gestures having both global and/or local hand motions. In the gesture recognition experiments discussed above, training and

[1]This work has been published in *IEEE Indicon 2006* (Refer item 15 in Page 199 for details)

TH-1870_02610203

testing samples are obtained by collecting gestures performed by 10 different signers at different instants of time and conditions. Each signer is provided manual instructions for performing different gestures. For a few signers, wider abnormal spatio-temporal and shape variations for some of the gestures are obtained, which affect the overall recognition rates. In the event of low illumination, self occlusion of fingers and palm affects the performance of the hand segmentation algorithm and in turn it greatly reduces the recognition accuracy.

4.4 Concluding Remarks

In this chapter, we have developed a universal hand gesture recognition system in the sense that the proposed gesture recognition system can handle different types of isolated dynamic gestures in a common platform. We considered scenarios where the time variation of the shape of the hand provides cues about the identity of the the activity performed by the hand. The proposed scheme is corroborated by experimental results for different hand gestures as well as static and dynamic signs from the native Indian sign language. The experimental results show very high recognition accuracy for all these cases. The system is also suitable for static hand gestures which may be considered as special cases of dynamic gestures.

The proposed recognizer may be suitable for recognizing hand gestures for HCI applications. However, it is to be noted that the proposed gesture recognizer cannot be considered as a complete sign language recognizer, as for complete recognition of sign language, information about other body parts *i.e.*, head, arm, facial expression, etc. are essential. The main motivation behind experiments on the signs of the native Indian sign language is to validate the proposed hand gesture recognition scheme.

TH-1870_02610203

Table 4.1: Experimental Results: Confusion matrix showing the gesture recognition rate using the proposed gesture recognition scheme.

No. of training samples per class: 50

No. of test samples per class: 150

Actual class labels	No. of test pattern assigned to different classes											Acc. rate	Err. rate	Rej rate
	Gesture No. 1	Gesture No. 2	Gesture No. 3	Gesture No. 4	Gesture No. 5	Gesture No. 6	Gesture No. 7	Gesture No. 8	Gesture No. 9	Gesture No. 10	Reject			
Gesture No. 1	140	0	0	0	0	0	10	0	0	0	0	93.3	6.7	0.0
Gesture No. 2	0	135	0	0	0	0	0	13	0	0	2	90.0	8.7	1.3
Gesture No. 3	0	0	144	0	0	0	0	0	0	0	6	96.0	0.0	4.0
Gesture No. 4	0	0	0	146	0	0	0	0	0	0	4	97.3	0.0	2.7
Gesture No. 5	0	0	0	0	137	0	0	0	0	10	3	91.3	6.7	2.0
Gesture No. 6	0	0	0	0	0	136	0	0	9	0	5	90.7	6.0	3.3
Gesture No. 7	4	0	0	0	0	0	141	0	0	0	5	94.0	2.7	3.3
Gesture No. 8	0	5	0	0	0	0	0	145	0	0	0	96.7	3.3	0.0
Gesture No. 9	0	0	0	0	0	5	0	0	145	0	0	96.7	3.3	0.0
Gesture No. 10	0	0	0	0	4	0	0	0	0	146	0	97.3	2.7	0.0
Overall Average												**94.3**	**4.0**	**1.7**

TH-1870_02610203

Table 4.2: Recognition rate for static hand signs taken from the native Indian sign language.

No. of training samples per class : 65 No. of test samples per class : 55

Gestures	Acc rate	Err rate	Rej rate
I Hand	98.18	1.8	0.0
II Hand	98.18	1.8	0.0
IV Hand	96.36	3.6	0.0
V Hand	94.54	3.6	1.8
Cartridge	98.18	1.8	0.0
Bowl	90.91	9.1	0.0
Bridge	90.91	9.1	0.0
Friend	94.54	5.4	0.0
High	98.18	0.0	1.8
Man	92.73	5.4	1.8
Mid day	94.54	1.8	3.6
Oath	96.36	3.6	0.0
Short	98.18	1.8	0.0
Drink	92.73	7.3	0.0
Up	92.73	7.3	0.0
Mirror	96.36	1.8	1.8
Average	**95.22**	**4.1**	**0.7**

TH-1870_02610203

Table 4.3: Recognition rate for isolated words in the native Indian sign language.

No. of training samples per class : 60 No. of test samples per class : 55

Gestures	Acc rate	Err rate	Rej rate
After	98.18	1.8	0.0
Alive	92.73	5.4	1.8
Alone	92.73	3.6	3.6
Grow	96.36	3.6	0.0
Arise	98.18	0.0	1.8
No	92.73	5.4	1.8
Call	90.91	3.6	5.4
Bless	98.18	0.0	1.8
Boy	96.36	1.8	1.8
Bring	96.36	3.6	0.0
Chief	92.73	1.8	5.4
Come	92.73	5.4	1.8
Defy	96.36	3.6	0.0
Depart	96.36	3.6	0.0
Distant	98.18	1.8	0.0
Fire	98.18	0.0	1.8
Peak	92.73	7.3	0.0
Wait	96.36	1.8	1.8
Fish	96.36	3.6	0.0
Jump	98.18	0.0	1.8
Give	96.36	3.6	0.0
Yes	98.18	1.8	0.0
Take	90.91	9.1	0.0
Shoot	96.36	1.8	1.8
Average	**95.53**	**3.1**	**1.4**

TH-1870_02610203

Chapter 5

Recognition of Continuous Dynamic Hand Gestures

One of the most important requirements for human computer interaction and sign language recognition is that natural gesturing should be supported by the recognition engine so that a user can interact with the system without any restrictions. Since a sequence of gestures is generally mixed with co-articulation and unintentional movements in between gestures, these non-gestural movements need to be eliminated from an input video sequence before each individual gesture in the sequence can be identified. In this chapter, we focus our attention to cope with this problem associated with continuous gesture recognition. This requires gesture spotting that distinguishes meaningful gestures from co-articulation and unintentional movements. In our method, we first segment the input video stream by detecting gesture boundaries at which the hand pauses for a while during gesturing. Next, every segment is checked for movement epenthesis and co-articulation via FSM matching or by using hand motion information. Thus, movement epenthesis and co-articulation phases are detected and eliminated from the sequence and we are left with a set of isolated gestures. Finally, we apply our earlier proposed recognition schemes to identify each individual gesture in the sequence. Our experimental results show that the proposed scheme is suitable for recognition of continuous gestures having different spatio-temporal behaviour.

TH-1870_02610203

5.1 Continuous Gesture Recognition

5.1.1 Problems in continuous gesture recognition

We have considered recognition of different types of isolated dynamic hand gestures in Chapters 2 to 4. However, in a real-world situation, a signer generally makes a series of gesture signs one after another. Therefore, a gesture recognition module can be of practical use only if it is capable of isolating each individual gesture in a continuous stream of gestures and apply appropriate recognition algorithm on each of them. For recognition of continuous gestures, there are two common problems that make the recognition task somewhat difficult and the recognition results may be sometimes inaccurate. These two problems are the problems of *gesture confusion* and *co-articulation*.

A continuous gesture sequence is generally a mixture of meaningful gestures and unintentional movements in between [Quek (1994), Pavlovic *et al.* (1997)]. Therefore, it is possible that an unintentional movement is confused as a meaningful gesture and vice versa. So, one very important issue in continuous gesture recognition research is to identify meaningful gestures in a continuous stream of body movements while removing all non-gestural motion components in the input video stream. This may be accomplished by spotting the exact instants when a gesture in the sequence starts and ends. This is called *gesture spotting*. Gesture spotting is essential for a recognition system to work continuously without any need for human intervention. Especially, it is possible for a user to interact with the recognition platform using natural gestures without any restriction only if gesture spotting is supported in a vision-based interface. By spotting gestures in a continuous video stream, we can automatically remove extra unintentional movements arising between gestures.

Another important issue in continuous gesture recognition is co-articulation which makes the extraction of gesture signs even harder in continuous hand movements. Co-articulation is the phenomenon in which one gesture influences the next in a temporal sequence [Vogler & Mextaxas (1997), Shamaie *et al.* (2001)]. In other words, when gestures are produced continuously, each gesture may be affected by the preceding gesture, and sometimes possibly by the gesture that follows it. In case of hand gestures, this happens due to non-gestural hand movement during transition from one gesture to the next. This is called *movement*

TH-1870_02610203

epenthesis. It is to be noted that movements due to co-articulation are not the same as unintentional movements in a continuous gesture sequence. Unintentional movements are generally erratic in nature and so cannot be categorized as such. Movement epenthesis occurs when the hand moves very fast from the last position in a gesture to the starting position in the following gesture. This implies that these movements are deliberate and are essential for connecting two gestures in a sequence. Nonetheless, co-articulation can also be eliminated by segmenting out each component gesture in a sequence precisely and accurately. Thus, the problem of continuous gesture recognition essentially boils down to precise gesture spotting.

Most vision-based gesture recognition systems developed so far assume that the input gestures are isolated or segmented. This assumption makes the recognition task easier, but at the same time it limits the naturalness of the interaction between the user and the system. In more natural settings, the gestures of interest are generally embedded in a continuous stream of motion, and their occurrence needs to be detected prior to recognition. This is the goal of gesture spotting. Subsequently, the recognizer attempts to classify the gesture as belonging to one of the predefined gesture classes. In the process, there are three possible types of errors: *insertion*, *deletion* and *substitution* [Lee & Kunii (1995), Starner *et al.* (1998)]. An insertion error occurs when the spotting algorithm reports a nonexistent gesture, while a deletion error occurs when the algorithm fails to detect a gesture. A substitution error occurs when a non-gesture is falsely classified as a gesture.

The basic idea in gesture spotting is to identify the starting and ending points of each gesture sign in a continuous gesture sequence. Once the gesture boundaries are known, a gesture in the sequence can be conveniently extracted out. However, the process of gesture boundary detection is not a trivial task because of the following reasons.

1. Gesture boundaries are subjective, *i.e.*, they vary from one signer to another.
2. Gesture boundaries are sequence dependent, *i.e.*, gesture segmentation is heavily influenced by higher level cognitive processes, and by the context surrounding each gesture [Kendon (1988)].
3. It is impossible to enumerate all gestures. Traditional computer vision approaches characterize a gesture as a series of poses. Clearly, given the virtually infinite number

TH-1870_02610203

of poses that the human body can assume, a generic model based approach to gesture segmentation is not viable.

5.1.2 Gesture spotting methods

Gesture spotting is a special case of the general pattern spotting problem where the goal is to find the pattern boundaries in long input signals like texts [Oudot *et al.* (2004)] , speeches [Rose (1996)], and image sequences [Morguet & Lang (1998)] including gesture video sequences. The different methods available for gesture spotting in literature may be categorized into three broad divisions as given below.

- **Direct approach:** All direct-approach algorithms first compute low-level motion parameters such as velocity, acceleration, and trajectory curvature [Kang *et al.* (2004)] or mid-level motion parameters such as human body activity [Kahol *et al.* (2004)], and then look for abrupt changes (*e.g.*, zero-crossings) in those parameters to detect gesture boundaries. An example of this type is the method used by Zhao *et.al.*, in which the local minima in the velocity profile serve as gesture boundaries [Zhao (2001)]. However, this approach generally fails to produce gestures that are consistent with human perception.

- **Indirect approach:** In this, gesture spotting is intertwined with the recognition process. The approach uses a state space model of gestures and the gesture boundaries are detected using recognition scores [Aggarwal & Cai (1997)]. First, a sequence of continuous gestures is represented as a series of static poses or states. Gesture segmentation is then achieved by identifying the start pose and the end pose of each gesture in the sequence. Starting from a state in the sequence, a gesture endpoint is detected when the recognition likelihood exceeds some threshold, and the gesture start point is determined by backtracking the optimal dynamic programming path. For example, Lee and Kim proposed a gesture spotting system that detects the end point of a gesture by using threshold model and finds a corresponding start point by searching the Viterbi path from the end point [Lee & Kim (1999)]. However, the method has a problem in that the system cannot report the detection of a gesture

TH-1870_02610203

immediately after the system reaches its end point. Moreover they used heuristic information, such as moving the hand out of the camera range. Such a heuristic is not very natural to humans. Nishimura and Oka also proposed a gesture spotting method using continuous dynamic programming which works on a frame-by-frame basis and matches a sequence of input feature vectors and standard patterns corresponding to each gesture [Nishimura & Oka (1996)]. However, this approach is limited to sets of gestures that do not contain any intermediate pose that resembles the start or end poses. Thus, in practical applications, this approach severely limits the number of gestures that can be segmented.

- **Combined approach:** The third category of gesture segmentation methods combines the indirect approach (state space-based approach) with direct approach (zero-crossing based approach). Vogler and Metaxas used context-dependent HMMs for segmentation and recognition of continuous gestures [Vogler & Mextaxas (1997)]. However context-dependent modelling has some inherent problems. First, it is linguistically implausible, because it fails to model *movement epenthesis* properly. An extra movement between the two signs are inserted when they are performed in sequence, which is not indicated by their lexical entries. This phenomenon is referred to as movement epenthesis [Liddell & Johnson (1989)]. Second, the number of states used in the HMM recognition network is of the order of $O(W^2)$, where W is the number of signs in the vocabulary which in general is very large. Later, they modified the algorithm by incorporating movement epenthesis for recognition of American Sign language (ASL) and showed some promising results by using parallel HMMs [Vogler & Mextaxas (1999)]. Wang *et. al.* used local minima in the acceleration profile as gesture boundaries to segment a gesture sequence into several motion alphabets or "characters" that are then combined using an HMM [Wang *et al.* (2001)]. When the motion alphabets derived from one motion sequence is used to segment a different motion sequence, the corresponding results are generally not satisfactory.

Though "gesture spotting" and "movement epenthesis detection" are two important and open research issues in continuous hand gesture recognition, not much vision based ap-

TH-1870_02610203

proaches for the same have been reported till date. The techniques developed so far for movement epenthesis detection are generally not useful for wide range of gesture vocabulary [Vogler & Mextaxas (1997), Morguet & Lang (1998), Shamaie & Sutherland (2001)]. This is because motion interpretation itself is an ill-posed problem in sense that a unique solution of the problem cannot be guaranteed [Wu & Huang (2001)]. For solution of such a problem, combination of cognitive science and psychological studies with computer vision has been suggested. Moreover, the existing algorithms do not address the problems associated with the recognition of different types of continuous hand gestures having different spatio-temporal behaviour or motion characteristics, *viz.*, gestures having only local motions, gestures having only global motions, gestures having both local and global motions, and gestures showing sequence of static hand poses (fluent finger spelling). In view of these, we propose to develop a scheme for gesture recognition that first identifies the nature of hand motion involved in a continuous gesture sequence and then applies a suitable spotting method incorporating cognitive and psychological aspects of recognition. In our method, we first segment the input video stream by detecting gesture boundaries at which the hand pauses for a while during gesturing. Next, every segment is checked for movement epenthesis via FSM matching or by using hand motion information. Thus, epenthesis phases are detected and eliminated from the sequence and we are left with a set of isolated gestures.

5.2 Proposed Gesture Spotting Methods

5.2.1 Gestural cues for continuous gesture segmentation

In order to segment out meaningful gestures from a continuous video stream and to eliminate any unintentional movement and/or co-articulation in between gestures, it is important to understand the nature and characteristics of gestures. Fortunately, some studies on human movement, including full-body gestures and hand gestures, had been done by researchers in the past and are available in literature. In this context, Efron's work [Efron (1941)] on how a gesture is performed was seminal to gesture analysis in psychology. He showed that

TH-1870_02610203

during gesturing humans use gestures as the building blocks for complex motions and divided the making of a hand gesture into three phases: preparation, stroke and retraction. He also showed that during gesture interpretation humans use visual cues to segment continuous motion sequences into discrete gestures and then perform recognition of each discrete gesture independently.

This pioneering work by Efron was later extended by Kendon. He showed that humans first segment a gesture, and then recognize it – a process analogous to segmentation of spoken words from a continuous stream of speech and then recognition of the segmented words by the human neuro-auditory system [Kendon (1988)]. He also made significant contribution by establishing that gestures are user specific, and that each human has a personal vocabulary of gestures that can be learned by speech-gesture correlations and acquaintance with that person. In another study, he defined a spectrum of gesturing behavior [Kendon (1990)]. According to him, on one end of this spectrum are the artificial and highly structured gestural languages, such as the American sign language (ASL), while on the other end are gestures that naturally and unconsciously co-occur with speech. In between these two extremes are the artificial but culturally shared emblems, such as the "thumbs-up" sign, the "V" sign and so on. He also described a hierarchy for gestures similar to Efron's work, a portion of which is shown in Figure 5.1. At the top level in the hierarchy is the gesture unit, a period of activity demarcated by the starting and returning of the hands to a rest position [Eisenstein & Davis (2005)]. This gesture unit is generally composed of a series of gesture signs or phrases. A gesture phrase is what is traditionally considered to be a single "gesture" – for example, pointing at something while talking about it, or tracing a path of motion. In the next level, every gesture phrase is decomposed into a set of hand strokes, which are considered to be the content-carrying parts of the gesture. In addition, there may also be a preparation phase which initiates the gesture, and possibly a retraction phase bringing the hand back to a pause state. Finally, the "hold" in the lowest level refers to a static positioning of the hand in gesture space, either before or after the stroke. These points have been already outlined in Chapter 1.

TH-1870_02610203

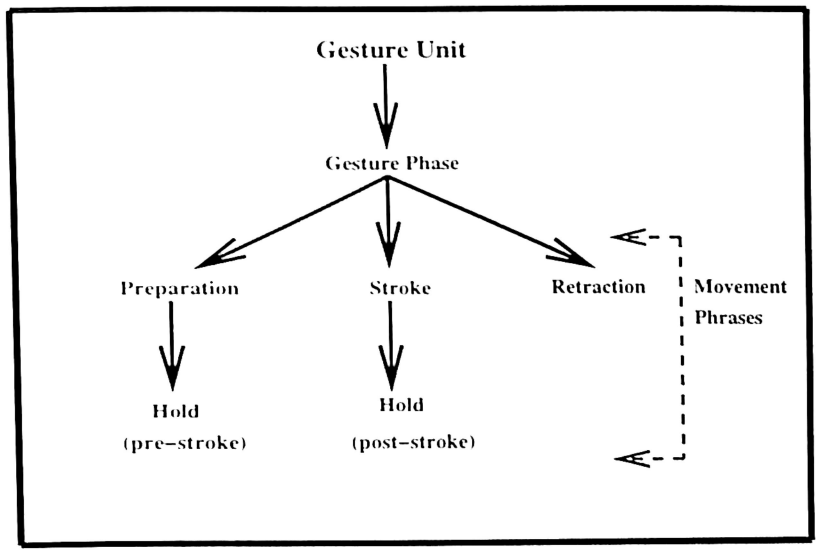


Figure 5.1: A hierarchy of gesture components [Eisenstein & Davis (2005)].

5.2.2 Gesture boundary detection

From the studies presented in the previous section, it can be deduced that generally a gesture starts and ends with the hand staying in a standstill position for a while. That is, a signer generally starts making a sign from a "pause" state and ends in a "pause" state even in case of continuous gesturing [Shriberg *et al.* (2000)]. Based on this idea, we propose to use the hand motion information for locating the boundary points of each individual gesture in a continuous stream of gestures. A boundary point is detected whenever the hand pauses during gesturing.

The first step in the proposed method for gesture spotting involves generating the video object plane (VOP) from each input frame in the continuous stream of video. A series of key VOPs are extracted from the generated VOPs using Algorithm 2.1, which also gives the duration of each key VOP in terms of the number of frames between each pair of key VOPs. A VOP model diagram showing a portion of a continuous gesture sequence is given in Figure 5.2. The diagram shows two gestures in the sequence connected with a movement epenthesis phase in between them. In the figure, $\mathrm{KVOP}_{m,n}$ represents the n^{th} key VOP in gesture number m and $\mathrm{T}_{m,n}$ is the corresponding time duration, expressed in terms of the number of video frames between $\mathrm{KVOP}_{m,n}$ and $\mathrm{KVOP}_{m,n+1}$

TH-1870_02610203

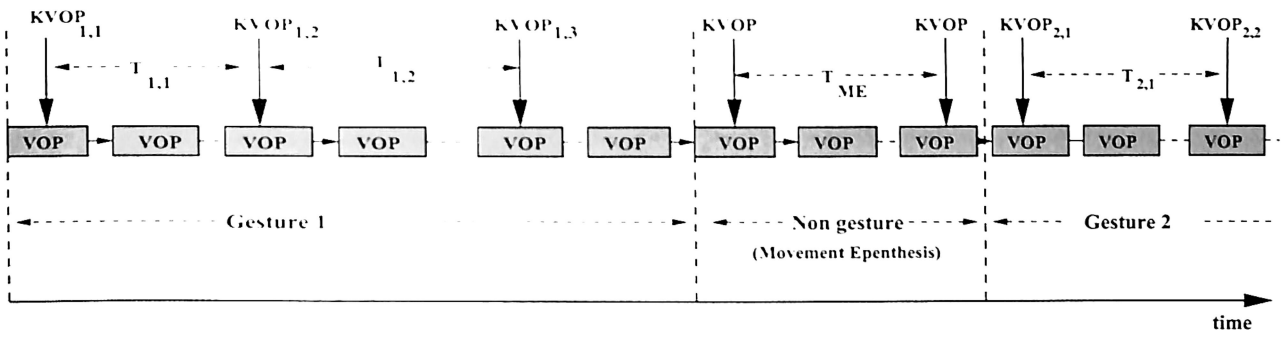


Figure 5.2: VOP model for a portion of a continuous gesture sequence showing movement epenthesis.

A key VOP in the sequence corresponds to a particular hand shape and/or position and the key VOP duration indicates the time for which the hand remains more or less fixed in that pose and position. This key VOP information is used in the proposed gesture spotting algorithm as well as for gesture recognition. For gesture spotting, a "pause" in the sequence is detected whenever the duration of a key VOP exceeds or at least equal to the minimum time for which the hand pauses at the starting or ending of a single gesture, as determined during the training session[1]. The use of key VOPs in gesture representation and recognition has been discussed earlier in Chapters 2 to 4.

In the proposed algorithm, we assume that the camera starts capturing the hand image sometime before the signer starts gesturing. Therefore, the first "pause" in the video indicates the starting of the first gesture in the sequence, as illustrated in Figure 5.3. The gesture may end when the next "pause" in the video stream is spotted. Assuming that there is always some non-gestural movement in between two gestures, the third and fourth "pauses" in the input video will indicate the starting and ending of the second gesture in the sequence respectively and so on. Thus, all the gestures in the input sequence can be selected out. However, the scheme will fail under certain situations, as listed below, and will result in incorrect spotting of gestures.

1. The assumption that there is always some non-gestural movement of hand in between two gestures may not be always true. For example, if the end position or pose of a

[1]This work has been published in *IEEE TENCON 2005* (Refer item 8 in Page 199 for details)

TH-1870_02610203

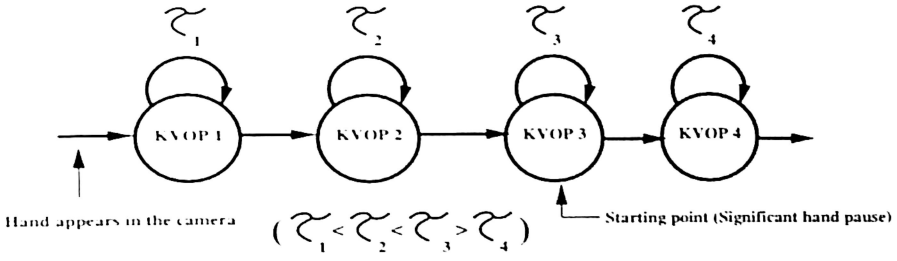


Figure 5.3: Gesture sequence starting point detection

gesture is same as the start position or pose of the next gesture then there is generally no extra movement in between these two gestures. In that case, the two gestures are adjoined to each other in the sequence with a common "pause" indicating end of the first gesture and start of the next gesture.

2. In the case of sequence of static hand poses (fluent finger spelling), there is generally no motion during the gesturing period while the hand may move in between two gesture poses due to movement epenthesis. That means, here a "pause" itself in the sequence corresponds to a gesture sign as if the start-point and end-point of the gesture have merged together.

3. In the case of gestures involving global hand motion, there may be some "pauses" within a single gesture. When the hand traverses in space, it makes one or more hand strokes to build up a complete gesture trajectory. Since a hand stroke generally starts from a "pause" and ends in a "pause", a multi-stroke gesture will contain some extra pauses in between.

In our research, we have visually analyzed a large number of real gesture trajectory patterns and identified twenty-four different gesture strokes to represent most of these trajectories. They are listed in Figure 5.4. Some examples of ideal gesture trajectories that are made up of one or more of these gesture strokes are shown in Figure 5.5. The first two examples are gestures having global motion only representing "One" and "Square", respectively. As we see in the figure, "One" is a single stroke gesture while "Square" is made up of four strokes. Therefore, the gesture "One" will start from a pause and will end in the next "pause". But, there will be three intermediate

TH-1870_02610203

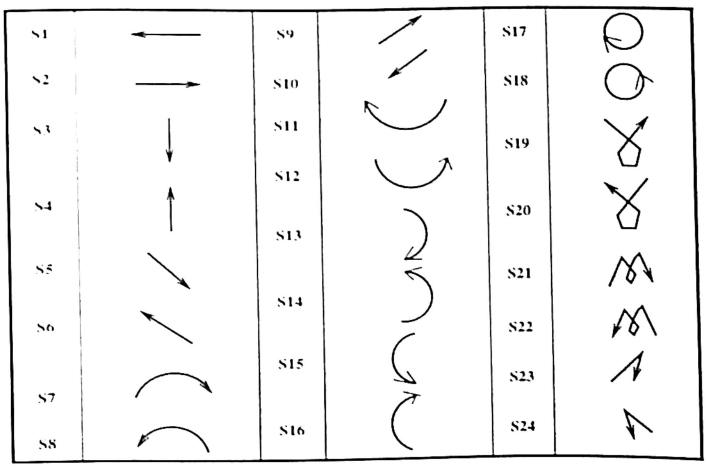


Figure 5.4: Basic hand strokes that are used in building different gesture trajectories.

"pauses" in the "Square" indicating gesture. The last example is the trajectory of a gesture composed of both local and global motions that represents a sentence in a sign language. Here each word in the sentence is signed by a single stroke associated with changing the hand pose. Hence, the number of strokes in the gesture is equal to the number of words in the sentence. Accordingly, for this example, there will be one extra pause in between two words.

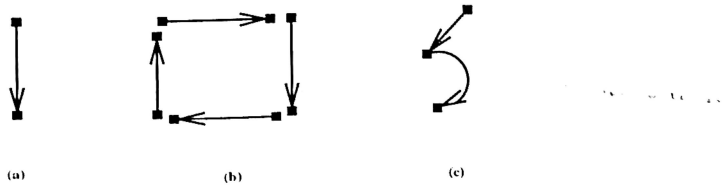


Figure 5.5: (a) Single hand stroke for a gesture (b) Multiple hand strokes for a gesture (c) Multiple strokes for a sentence of sign language.

Hence, it is not always possible to have reliable spotting of gestures only by detecting pauses in an input video. In view of this, we propose to check the nature of hand movement in every video segment in between two "pauses" in the input stream. Assuming that there is no unintentional movement other than movement epenthesis in between gestures, the proposed

TH-1870_02610203

gesture spotting method requires to determine the occurrence of movement epenthesis in the sequence. This is done through FSM matching and/or using motion features, as explained in the next section.

5.2.3 Movement epenthesis detection

In order to detect the occurrence of movement epenthesis between two "pauses" in a continuous video stream, we look for some characteristic features to decide whether it is a movement epenthesis phase or not. Movement epenthesis occurs when the gesturing hand moves fast from the last pose (or position) of a gesture to the first pose (or position) of the next gesture in the sequence. That means, in gestures with local motions only, movement epenthesis is characterized by fast change in hand shape (pose) while in gestures having global motion it is indicated by fast movement of the hand in space.

Movement epenthesis detection in continuous gestures with local motion only

Following two steps are proposed for movement epenthesis detection for gestures having only local hand motions.

Step 1: We assume that after completing a gesture, the signer holds his hand for sometime in the last signed pose in the gesture and then quickly moves it to the starting pose of the next gesture in the sequence. The signer does this by bending his fingers and/or moving his palm in a very short span of time while holding the hand more or less fixed at one position in space. That means, the movement epenthesis phase is also made up of local hand motions only and hence may be represented by an FSM model. Now since there is fast change in hand shape during movement epenthesis compared to that during a gesturing phase, the time duration associated with each state of an FSM representing movement epenthesis will be generally very small than that for an FSM representing a gesture. Therefore, movement epenthesis in continuous gesturing may be detected if the key VOP durations in between two "pauses" are below a certain threshold. The value of this threshold may be decided during the training session.

Step 2: The movement epenthesis detected in Step 1 can be verified by representing the

TH-1870_02610203

KVOPs in the detected movement epenthesis phase by an FSM. Following the procedure described in Chapter 2, the input video sequence is represented by an FSM and is matched to all the prototype FSMs contained in the database, each prototype FSM representing a local motion gesture in the gesture vocabulary. If no match is obtained then movement epenthesis is verified.

Movement epenthesis detection in continuous gestures having global motion

In the case of gestures having global motion only or gestures having both global and local motions, the gesturing hand traverses in space to form a gesture trajectory. After a gesture trajectory is completed, the hand pauses for a while and then moves with very high velocity to the starting position of the next trajectory. After that, the hand again pauses for a while before starting the next trajectory. Based on this, we now propose to detect movement epenthesis by observing the motion of the hand between two "pauses" in the input hand motion video.

As discussed earlier, a gesture phase can be divided into three motion stages – preparation, stroke and retraction, in addition to the starting and ending "pauses". On the other hand, a movement epenthesis phase starts from a pause, makes a fast hand stroke and finally ends up in another pause. Therefore, it is possible to distinguish a movement epenthesis from a gesture stroke if we can determine the behaviour of the gesturing hand in between two "pauses". For this, we compute two motion parameters, *viz.*, velocity and acceleration, at every key VOP instant and decide the nature of hand movement at that instant. The proposed scheme for movement epenthesis detection in the continuous gestures with global hand motions consists of the following two steps.

Step 1: Movement epenthesis detection by motion features

As discussed above, two motion features are used in our system for detecting movement epenthesis phase.

Acceleration feature for movement epenthesis detection:

During movement epenthesis the speed of the hand increases to a very high value from almost zero value and then abruptly comes down to almost zero as illustrated in Figure 5.6.

TH-1870_02610203

That means, the hand moves with very high acceleration (positive or negative) during the movement epenthesis phase. On the other hand, during gesturing the speed of the hand gradually increases from a pause, may remain constant for sometime and then gradually comes down to almost zero. Therefore, acceleration feature may be a good measure to check for movement epenthesis, as mentioned earlier in Chapter 3.

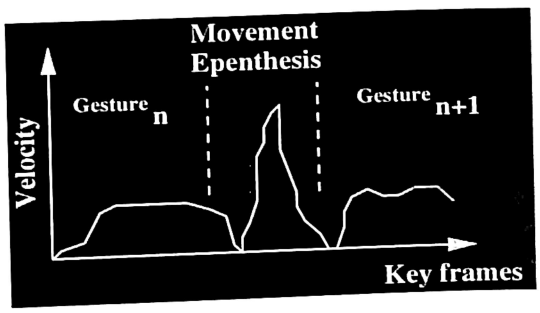


Figure 5.6: Example of a typical velocity plot for two sequentially connected gestures having global motion in a continuous video stream.

Velocity feature for movement epenthesis detection:

The speed of the hand is generally very high while making a stroke. But, that during the preparation and retraction stages is generally very small. That means, the average velocity of the hand during movement epenthesis is generally very large compared to that during a gesturing phase. Hence, the measure of velocity may serve as an additional feature to detect movement epenthesis.

From the analysis of data corresponding to the velocity and the acceleration for different gestures, it is seen that the values of these two parameters are not unique and precise for different samples of a same gesture. The discrimination of imprecise data based on these two parameters can be done by using the Fuzzy-rule based method. We now propose a fuzzy-rule based method to discriminate between a movement epenthesis phase and a gesturing phase using these two motion parameters.

Fuzzy method for movement epenthesis detection:

Fuzzy logic is a multi-valued logic that allows an *input* to have its belongingness to a

TH-1870_02610203

fuzzy set in terms of degrees of *membership* in the interval [0, 1] [Zadeh (1988)]. The fuzzy membership function is generally defined in a such way that it has membership grade 1 (full certainty) at a particular value or over a range, and then the membership grade gradually decreases from 1 to 0 as the input value deviates away. The shape of the membership function is at user's choice depending on the problem at hand. Some commonly used examples are the Gaussian, the triangular and the trapezoidal membership functions [Bezdek (1981)]. The triangular and the trapezoidal membership functions are chosen in most applications because of their simplicity. Fuzzy logic operations like $min(A, B)$ and $max(A, B)$ are applied on a fuzzy set to construct fuzzy rules in the form of *if-then* statements. The *max* function is used for the fuzzy OR operation, while truncation or AND operation is done with the *min* function. The process of generating membership values for a fuzzy variable using a set of input membership functions is known as *fuzzification*. For applying the fuzzy rules, the fuzzy set is to be decided first. This is generally done through experiments and observations.

A fuzzy-logic based method excels in problem solving with imprecise data by trading off between significance and precision similar to the reasoning process in humans. It is tolerant of imprecise data [Zadeh (1988)]. Such a method promises to be suitable for discriminating between a movement epenthesis phase and a gesturing phase using the velocity and the acceleration as the inputs.

We have formulated a scheme to measure the motion behavior in terms of some fuzzy sets and rules to determine whether a particular motion is a gesture phase or movement epenthesis[1]. By observing different gesture samples, we first define four fuzzy sets to denote the different ranges of speed. They are 'Zero' (ZO), 'Positive Small' (PS), 'Positive Medium' (PM) and 'Positive Large' (PL). The corresponding fuzzy membership functions are constructed by experimentation and plotted in Figure 5.7. Similarly, we define five fuzzy sets to quantify change in speed in terms of some fuzzy measures – 'Negative Medium' (NM), 'Negative Small' (NS), 'Zero' (ZO), 'Positive Small' (PS) and 'Positive Medium' (PM); the corresponding membership functions are shown in Figure 5.8. For computing the absolute values of speed and the change in speed, we calculate the velocity and the ac-

[1]This work has been published in *Springer-Verlag 2006* (Refer item 2 in Page 198 for details)

TH-1870_02610203

celeration using equations 3.12 and 3.13 given in Chapter 3. These two motion parameters are subsequently fuzzified and the motion stage through which the hand is undergoing is determined using some fuzzy rules, as stated below.

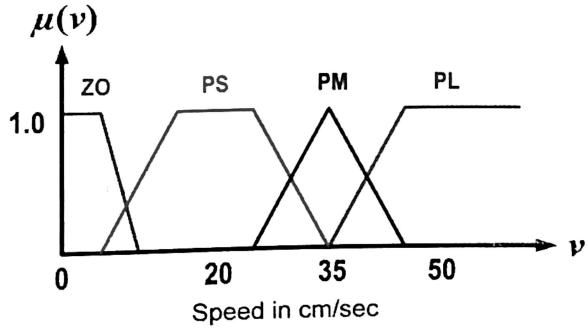


Figure 5.7: Fuzzy membership functions defining different ranges of speed.

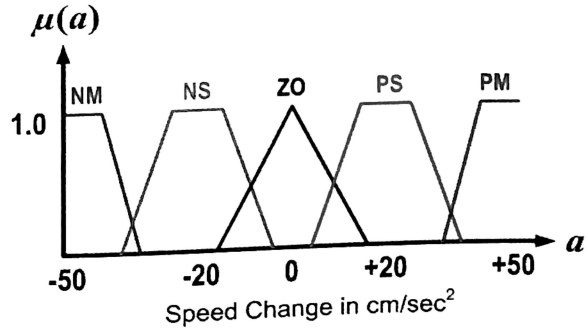


Figure 5.8: Fuzzy membership functions defining different ranges of change in speed.

From a large set of real gestures we observe that the speed of the hand generally lies within a certain range at every stage in a gesture or movement epenthesis. Similar is the case for change in speed. For example, during the "Preparation" stage the speed is generally 'Positive Small' and the change in speed is either 'Positive Small' or 'Negative Small'. We can write this in the form of a fuzzy rule as

- "IF the speed is Positive Small AND the change in speed is Positive Small OR Negative Small, THEN the hand is in Preparation stage".

TH-1870_02610203

Accordingly, if the hand is moving at a speed v and the change in speed is a then, the degree of certainty by which we can say that the hand is in "Preparation" stage is given as

$$\mu_{\text{Prep}}(v, a) = \min\left[\mu_{\text{PS}}(v), \max\left[\mu_{\text{PS}}(a), \mu_{\text{NS}}(a)\right]\right] \tag{5.1}$$

The IF-THEN fuzzy rules for all the other three motion phases can be derived in a similar manner and using Table 5.1 that gives the fuzzy sets for the two motion parameters corresponding to the different stages of hand motion. The fuzzy rule for each motion phase is obtained by the INTERSECTION of the UNIONS of the fuzzy sets for each individual input. Using all these fuzzy rules we can compute the degrees of confidence that the hand is doing "Pause", "Preparation", "Stroke" and "Retraction" at a given instant of time. Finally, we decide for the motion stage that has the maximum degree of confidence. In doing so, we are able to find the sequence of hand motion in an input stream of hand gesture video. If the extracted motion pattern follows the motion phases of Table 5.1 in order, the video segment is classified as a gesture, otherwise it is labelled as movement epenthesis.

Table 5.1: Gesture motion phases and corresponding motion parameters

Motion phase	Speed	Change in Speed
Pause	ZO	ZO, PS, NS
Preparation	PS	PS, NS
Stroke	PL, PM	ZO, PS, NS, PM, NM
Retraction	PS	NS

Step 2: Verification of movement epenthesis using trajectory shape

The process of movement epenthesis detection can be made more reliable and accurate by considering the hand trajectory shape for verification. It is observed that during movement epenthesis the hand generally moves in a straight path. So, a motion phase is not a movement epenthesis if the trajectory is not a straight line. Note that a straight-line trajectory does not always indicate movement epenthesis. It may also represent a valid gesture stroke. The Step I discriminates between movement epenthesis and such a gesture stroke.

TH-1870_02610203

A continuous gesture recognition scheme incorporating appropriate method for movement epenthesis detection is developed in the next section.

5.3 Proposed Algorithm for Gesture Recognition

5.3.1 Types of continuous gesture sequences

As said earlier, an isolated dynamic hand gesture may be composed of local motions only, global motion only, or combination of both local and global motions. Therefore, in principle, a continuous gesture sequence may be a mix of any of these three types of gestures. That is, the sequence may have one type of gestures only, or a mix of any two types of gestures, or a mix of all three types of gestures. Again continuous gestures with local hand motions only may be of two types. In one type, the individual gestures in the sequence are dynamic gestures having only local motion. The other kind includes gesture sequences that are composed of static hand poses only. It is to be noted that for the latter kind, though each individual gesture sign is a static pose, the gesture sequence as a whole is never static due to hand movement from one sign to the next in the sequence. In this work, we consider four types of continuous hand gestures that may be useful in HCI related applications. These are:

Type I gestures that are composed of static hand pose gestures only.

Type II gestures that are composed of hand gestures having local hand motions only.

Type III gestures that are composed of hand gestures having global motion only.

Type IV gestures that are composed of gestures having both local and global motions.

Further, we make the following assumptions on the choice of gestures in the vocabulary.

- A gesture as a whole cannot be the starting part of another gesture in the vocabulary.
- None of the gestures suffers from *intermediate movement epenthesis*. Intermediate movement epenthesis occurs when there is some extra movement in between two strokes of the same gesture. For example, a "Plus" representing global motion gesture is formed by one vertical stroke and one horizontal stroke intersecting each other.

TH-1870_02610203

Here the two strokes will be connected by an extra stroke joining the bottom point of the vertical stroke (first stroke) to the leftmost point on the horizontal stroke following it. Some more examples of intermediate movement epenthesis gestures are given in Figure 5.9.

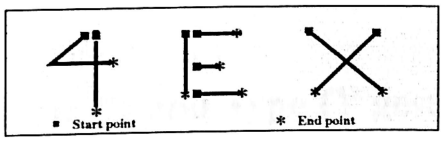


Figure 5.9: Examples of gestures having intermediate movement epenthesis

5.3.2 Gesture type detection

The proposed algorithm for continuous hand gesture recognition starts with VOP generation, key VOP extraction and "pause" detection, as explained earlier. For this, a few key VOPs following the first "pause" in the sequence is buffered. The input sequence is classified as only local motion gestures (Type I and Type II) or as global motion gestures (Type III and Type IV) based on the positions of the VOP centroids in these buffered key VOPs. The method for the same has been presented earlier in Chapter 4. The next step involves gesture recognition as described below.

5.3.3 Recognition in Type I and Type II gestures

As already described in this chapter, interpretation of a continuous stream of gestures requires to separate out every individual gesture in the input sequence followed by identification of the same. For the purpose of individual gesture identification in Type I and Type II gestures, all key VOPs in between every pair of "pauses", both inclusive, are taken and a finite state machine is constructed. This FSM is then matched with all prototype FSMs contained in our local motion gesture vocabulary. The algorithm for FSM matching has been described in Chapter 2. The concerned video segment is identified as the gesture whose prototype FSM matches with the input. As explained earlier, if no match is found then this segment corresponds to a movement epenthesis phase and is discarded. After this, we move on to the next segment and the process continues as long as the signer keeps on signing in front of the camera.

TH-1870_02610203

A special situation may arise when all the segments are identified as movement epenthesis phases. This corresponds to the Type I gestures. For this, the FSMs corresponding to static gestures (FSM with infinite time duration) are considered for matching. Recognition is accomplished by matching the state corresponding to each key VOP with the states of the above FSMs.

5.3.4 Recognition in Type III and Type IV gestures

In these types of gestures, every video segment between two "pauses" corresponds to either a gesture stroke in space or a movement epenthesis phase. Therefore, as a first step towards recognition, it is required to check whether an input segment is a movement epenthesis or a gesture stroke. It is accomplished by fuzzy rule-based method as described in the previous section.

Consider the first stroke between two pauses and apply the check for the movement epenthesis to it. If it is a movement epenthesis then it is discarded and we move on to the next segment. If it is a stroke then we check whether it is a complete gesture trajectory or not by comparing it to all the prototype trajectories contained in the gesture vocabulary database. Otherwise, we move on to the next segment and repeat the same procedure till a valid gesture trajectory is obtained by connecting all these individual hand strokes in a sequential manner.

Once a complete gesture trajectory is obtained, we proceed for FSM based matching. FSM matching will be successful if the input is of Type IV. Else, feature based trajectory matching is applied for recognition of Type III gestures. The algorithm for gesture recognition using trajectory and/or FSM matching has been presented earlier in Chapter 4.

5.4 Experimental Results

A number of experiments were performed to verify the proposed continuous gesture recognition algorithm. The same camera and system setups of the experiments of earlier chapters were used. During all these experiments, signers are initially acquainted with different gestures and asked to perform all the gestures according to the guidelines fixed before.

TH-1870_02610203

The VOPs were generated and the key VOPs were extracted using methods described in Chapter 2. For determining types of gesture sequences, we buffered 5 consecutive key VOPs just after the hand appears before the camera. It is seen that, initially the hand requires some time for stabilization against unintentional movement and hand trembling for making a gesture. We experimentally found that selection of 5 initial key VOPs is computationally sufficient enough to take a decision regarding type of motions present in a particular gesture sequence.

Five sets of experiments are performed on different types of continuous dynamic hand gesture sequences. These experiments are listed as follows.

- **Experiments with continuous gestures of Type I:**
  In this, we considered five hand poses/shapes as shown in the Figure 5.10. These five hand poses are performed one after another in different orders. By doing so, 25 sequences of hand poses are obtained for experimentation. Five signers are asked to repeat each sequence for 10 times. So we have altogether 1250 different sequences. Out of 1250 gesture sequences, 450 sequences were used for training. Table 5.2 shows the recognition rates for each of the selected 25 continuous gesture sequences. Relatively low recognition rates are found for those continuous gesture sequences that have signs 'two' and 'friend' consecutive to each other. It may be due to the nearly identical hand shapes for these two gestures. The overall recognition rate for 800 test sequences are found as 89.8%, which demonstrates the effectiveness of the proposed system.



Figure 5.10: Static hand poses 'One', 'Two', 'IV hand', 'Cartridge' and 'Friend' respectively used in our 1st experiment.

TH-1870_02610203

Table 5.2: Gesture recognition rates for Type I gesture sequence.

No. of training samples per class : 18 No. of test samples per class : 32

Continuous gestures	Recognition rates
One – Two – IV hand – Cartridge – Friend	96.87
One – Two – IV hand – Friend – Cartridge	96.87
One – Two – Friend – Cartridge – IV hand	84.37
One – Two – Friend – IV hand – Cartridge	87.50
One – Two – Cartridge – Friend – IV hand	96.87
One – Two – Cartridge – IV hand – Friend	96.87
One – IV hand – Two – Cartridge – Friend	90.62
One – IV hand – Two – Friend – Cartridge	90.62
One – IV hand – Friend – Two – Cartridge	87.50
One – IV hand – Friend – Cartridge – Two	96.87
One – IV hand – Cartridge – Friend – Two	84.37
One – IV hand – Cartridge – Two – Friend	81.25
One – Cartridge – IV hand – Two – Friend	84.37
One – Cartridge – IV hand – Friend – Two	84.37
One – Cartridge – Friend – Two – IV hand	84.37
One – Cartridge – Friend – IV hand – Two	84.37
One – Cartridge – Two – IV hand – Friend	96.87
One – Cartridge – Two – Friend – IV hand	81.25
One – Friend – Cartridge – Two – IV hand	96.87
One – Friend – Cartridge – IV hand – Two	96.87
One – Friend – IV hand – Two – Cartridge	90.62
One – Friend – IV hand – Cartridge – Two	90.62
One – Friend – Two – IV hand – Cartridge	84.37
One – Friend – Two – Cartridge – IV hand	90.62
Friend – One – Two – Cartridge – IV hand	90.62
Average	**89.80**

TH-1870_02610203

- **Experiments with continuous gestures of Type II:**

  In a second set of experiments, we considered some sequences of continuous gesturing with local hand motions only. The gesture vocabulary used in this case contained the same ten different gesture signs that we used in our experiments on isolated local motion gesture recognition in Chapter 2. The gestures in the vocabulary were divided into two groups arbitrarily each consisting of 5 gestures. A combination of 5 gestures in a group was considered to obtain a continuous gesture sequence. One such combination was repeated 25 times by 5 signers. Thus a total of 1250 samples of continuous gesture sequences are obtained for experimentation. The same partitioning as in the experiment with continuous gesture of Type I was used to divide the gesture sequences into training and testing sequences. A large number of these sequences contain intermediate movement epenthesis. Figure 5.11 shows a portion of a continuous gesture sequence where some selected frames are shown. The transition from "close to one" gesture to "close to two" gesture occurs with a movement epenthesis phase in between. We used the same FSM database that we obtained through training earlier in Chapter 2. In our experiments, we achieved an overall recognition accuracy of 90.9%, with highest and lowest recognition rates of 96.87% and 81.25% respectively. This high recognition rate confirms that the proposed algorithm for gesture spotting and movement epenthesis detection was indeed effective in segmenting out meaningful individual gestures in the input sequences accurately and precisely.

- **Experiments with continuous gestures of Type III:**

  In a third set of experiments, we considered trajectory patterns indicating "One", "Two", "Five", "Seven" and "Three", as shown in the first row of Figure 5.12(a). These gesture signs were performed one after another in different orders to build up different sequences of continuous hand motion gestures with global motion only. As we observe, each gesture starts from some point at top of the frame and ends somewhere at the bottom. Thus, we have a movement epenthesis in between two gestures in a sequence. In this case, we used the fuzzy-rule based method to detect movement epenthesis hand strokes as discussed in Section 5.2.3. Figure 5.12(b) shows

TH-1870_02610203

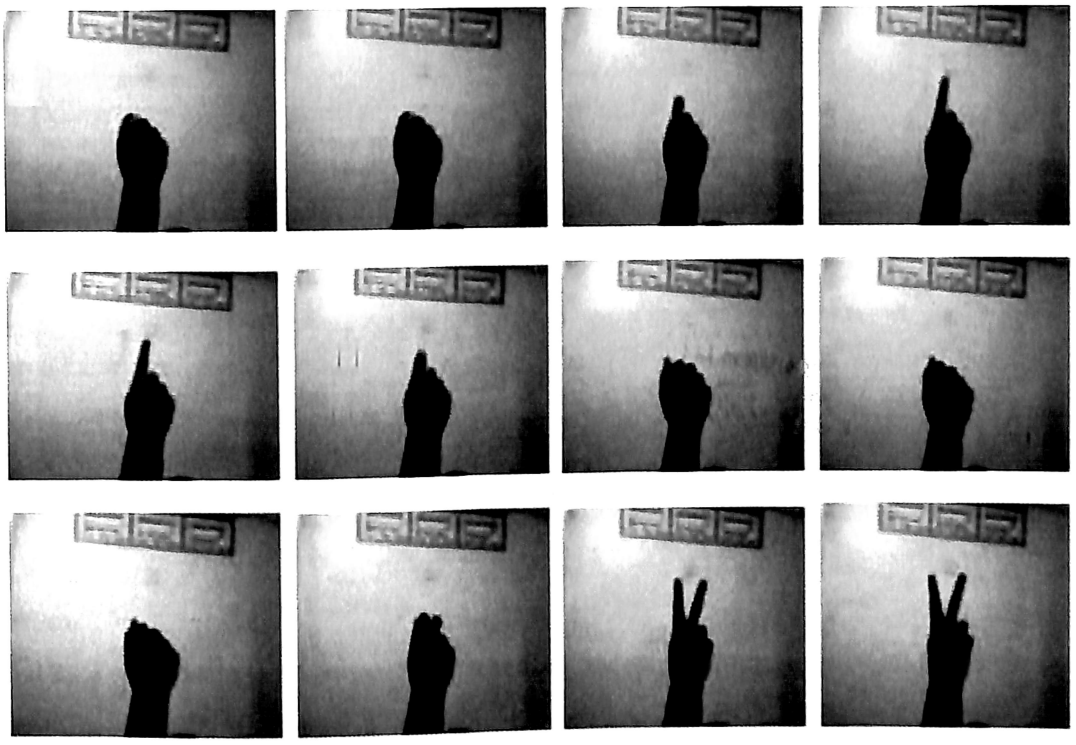


Figure 5.11: A portion of a continuous gesture sequence showing transition from one gesture to the next used in our 2nd experiment.

another set of gesture trajectories, *viz.*, "Square", "Circle", "Diamond", "Triangle" and "W". These gestures were used in the fourth set of experiments. In this set of experiments, the starting and ending of all gestures in a sequence are in the vicinity of each other. Here the gestures are generally connected to each other without any movement epenthesis in between. Figure 5.13(a) shows a sample of the continuous gesture trajectory where the gestures do not have any movement epenthesis in between. Figure 5.13(b) shows a a sample of the continuous gesture trajectory connected by movement epenthesis in between. The segmented out gesture trajectories from the gesture sequence samples in Figure 5.13(a) and Figure 5.13(b) are shown in Figure 5.13(c). In this figure, the first row shows the segmentation result corresponding to Figure 5.13(a) and the second row shows that corresponding to Figure 5.13(b). These demonstrate the efficacy of the gesture spotting method.

TH-1870_02610203

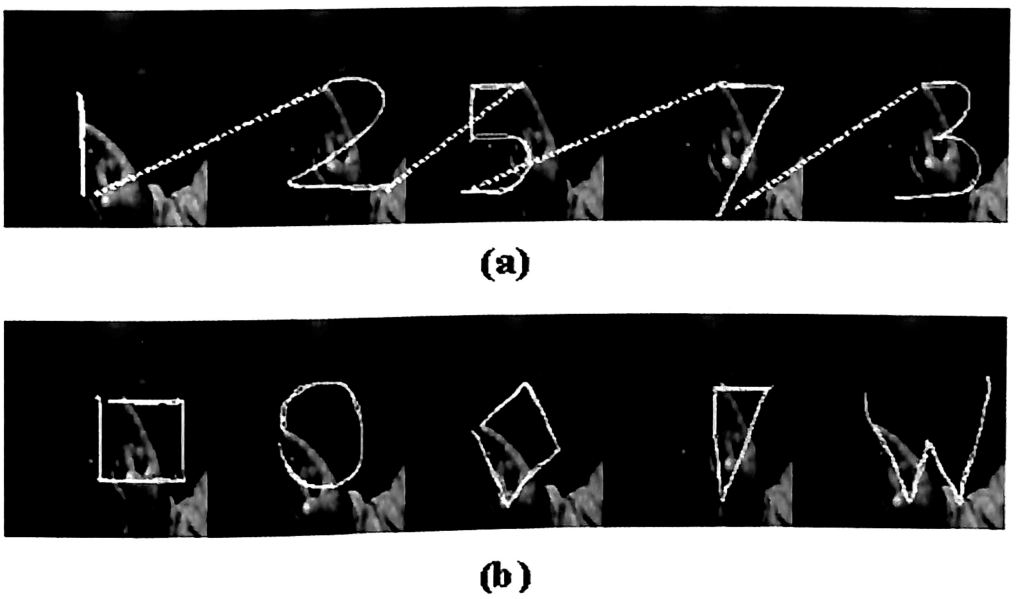


Figure 5.12: Trajectories of the gestures used in our experiments: (a) Gestures that produce movement epenthesis, used in our 3rd set of experiments, (b) Gestures that do not produce movement epenthesis, used in our 4th set of experiments.

In the third set of experiments, we used hand motion information to discriminate the movement epenthesis phases in the input gesture sequences. Subsequently, all individual gesture patterns in the sequences were segmented out and were identified with an overall accuracy of 94.4%. Table 5.3 shows the classification accuracy in case of the selected gesture vocabulary. This demonstrates the efficiency of our proposed method for movement epenthesis detection and subsequent trajectory guided recognition. We also observed that the acceleration of the hand was significantly high during movement epenthesis compared to that during the gesturing phase. Table 5.4 gives the values of average acceleration during the gesturing and movement epenthesis phases in case of two sequences used in our experiments. Finally, in our fourth set of experiments, we obtained an overall recognition rate of 90.4%. Table 5.5 shows the recognition rates for each of the selected 25 continuous gesture sequences. This shows that our gesture recognition system is capable of identifying gesture strokes and trajectory patterns with very high accuracy. The high recognition rate observed

TH-1870_02610203

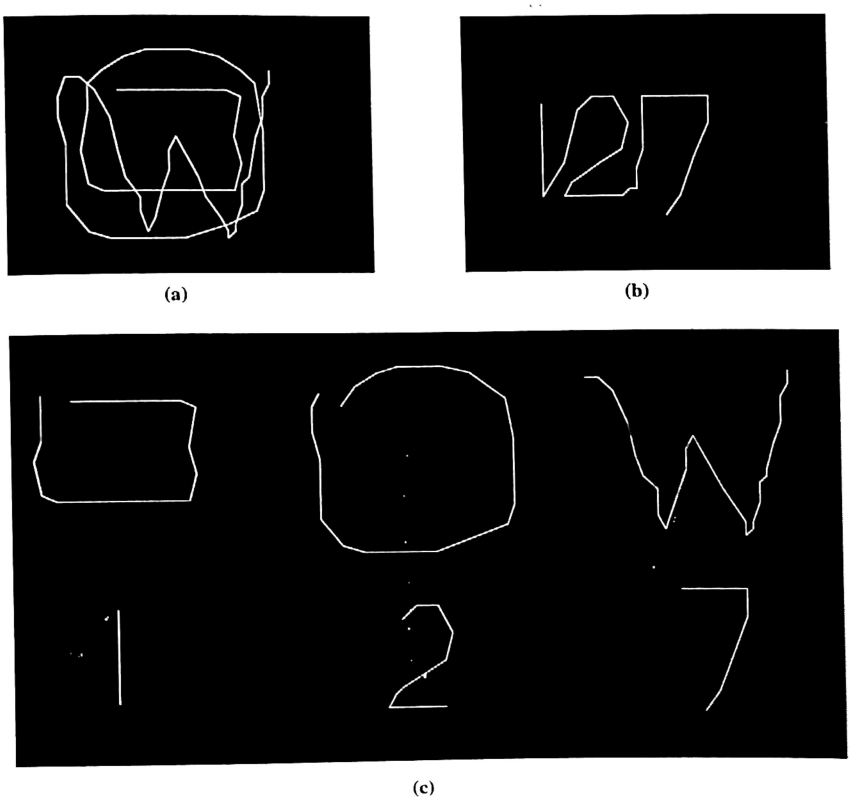


Figure 5.13: (a) A continuous motion trajectory without movement epenthesis (b) A continuous motion trajectory with movement epenthesis (c) Segmented out trajectories.

in the third set of experiments compared to the fourth set of experiments may attributed to the exact gesture spotting through the detection of movement epenthesis. Moreover, it is seen during experimentation that most errors come from the failure of hand extraction that distorts the hand trajectory data.

- **Experiments with continuous gestures of Type IV:**

  In a fifth set of experiments, we considered sequences of continuous gesturing with both local and global hand motions. The gesture vocabulary used in this case contained the first five gesture signs [Figures 4.8 (a)-(e)] out of ten used in the experiments on isolated gesture recognition in Chapter 4. The continuous gesture sequences were formed by performing these gesture signs in different orders. In this case also, 25 different gesture sequences were obtained. Each sequence was performed by 5

TH-1870_02610203

Table 5.3: Gesture recognition rates for Type III (3rd set of experiments) gesture sequence.

No. of training samples per class : 10 No. of test samples per class : 15

Continuous gestures	Recognition rates
One – Two – Five – Seven – Three	100.00
One – Two – Five – Three – Seven	93.33
One – Two – Three – Seven – Five	93.33
One – Two – Three – Five – Seven	93.33
One – Two – Seven – Three – Five	100.00
One – Two – Seven – Five – Three	100.00
One – Seven – Five – Three – Two	86.67
One – Seven – Five – Two – Three	86.67
One – Seven – Two – Three – Five	93.33
One – Seven – Two – Five – Three	93.33
One – Seven – Three – Two – Five	93.33
One – Seven – Three – Five – Two	100.00
One – Three – Five – Two – Seven	100.00
One – Three – Five – Seven – Two	93.33
One – Three – Seven – Two – Five	100.00
One – Three – Seven – Five – Two	86.67
One – Three – Two – Seven – Five	93.33
One – Three – Two – Five – Seven	93.33
One – Five – Three – Two – Seven	100.00
One – Five – Three – Seven – Two	100.00
One – Five – Seven – Two – Three	100.00
One – Five – Seven – Three – Two	93.33
One – Five – Two – Seven – Three	86.67
One – Five – Two – Three – Seven	86.67
Two – Three – Five – Seven – One	93.33
Average	**94.40**

TH-1870_02610203

Table 5.4: Average acceleration in cm/sec^2 in continuous gesturing

Gesture Sequences	1st Gesture phase	Movement epenthesis phase	2nd Gesture phase
"One" → "Two"	0.342	0.783	0.338
"Five" → "One"	0.257	0.845	0.276

different signers for 5 times each, thereby resulting 625 number of continuous gesture sequences. Out of 625 gesture sequences, 250 sequences were used for training. The classification results on the test data using the proposed method of recognition are shown in Table 5.6. The overall recognition rate for 375 test sequences are found as 92.3%, which is higher than the recognition rate for the continuous gestures with local motions only.

The details of separation of total data into two sets *viz.*, training set and testing set for all these experiments along with corresponding overall recognition rates are shown in Table 5.7. Recognition rates for all these cases are satisfactory in the context of vision-based continuous dynamic hand gesture recognition. These results suggest the possible recognition of continuous hand gestures through vision based technique.

5.5 Concluding Remarks

Movement epenthesis and co-articulation detection is one of the main challenges in continuous gesture recognition. It is difficult to distinguish between a valid gesture phase and the movement epenthesis phase only from the motion information. That is why, not many vision-based approaches for estimating movement epenthesis have been reported in the literature till date. Most of the existing algorithms use the glove environment and are successful only on some specific gesture vocabularies. A generalized approach is needed to handle all kinds of gestures performed in different contexts in a common platform. The proposed

TH-1870_02610203

Table 5.5: Gesture recognition rates for Type III (4^{th} set of experiments) gesture sequence.

No. of training samples per class : 10 No. of test samples per class : 15

Continuous gestures	Recognition rates
Square – Circle – Diamond – Triangle – W	86.67
Square – Circle – Diamond – W – Triangle	86.67
Square – Circle – W – Triangle – Diamond	86.67
Square – Circle – W – Diamond – Triangle	93.33
Square – Circle – Triangle – W – Diamond	93.33
Square – Circle – Triangle – Diamond – W	93.33
Square – Triangle – Circle – Diamond – W	93.33
Square – Triangle – Circle – W – Diamond	80.00
Square – Triangle – W – Diamond – Circle	80.00
Square – Triangle – W – Circle – Diamond	86.67
Square – Triangle – Diamond – W – Circle	86.67
Square – Triangle – Diamond – Circle – W	86.67
Square – Diamond – Circle – W – Triangle	93.33
Square – Diamond – Circle – Triangle – W	93.33
Square – Diamond – Triangle – W – Circle	93.33
Square – Diamond – Triangle – Circle – W	86.67
Square – Diamond – W – Triangle – Circle	86.67
Square – Diamond – W – Circle – Triangle	86.67
Square – W – Circle – Triangle – Diamond	100.00
Square – W – Circle – Diamond – Triangle	100.00
Square – W – Diamond – Triangle – Circle	93.33
Square – W – Diamond – Circle – Triangle	93.33
Square – W – Triangle – Circle – Diamond	100.00
Square – W – Triangle – Diamond – Circle	93.33
Triangle – Diamond – Circle – Square – W	86.67
Average	**90.40**

TH-1870_02610203

Table 5.6: Gesture recognition rates for Type IV gesture sequence.

No. of training samples per class : 10 No. of test samples per class : 15

Continuous gestures	Recognition rates
Gesture 1 – Gesture 2 – Gesture 3 – Gesture 4 – Gesture 5	93.33
Gesture 1 – Gesture 2 – Gesture 3 – Gesture 5 – Gesture 4	100.00
Gesture 1 – Gesture 2 – Gesture 5 – Gesture 4 – Gesture 3	86.67
Gesture 1 – Gesture 2 – Gesture 5 – Gesture 3 – Gesture 4	86.67
Gesture 1 – Gesture 2 – Gesture 4 – Gesture 3 – Gesture 5	86.67
Gesture 1 – Gesture 2 – Gesture 4 – Gesture 5 – Gesture 3	93.33
Gesture 1 – Gesture 3 – Gesture 4 – Gesture 5 – Gesture 2	80.00
Gesture 1 – Gesture 3 – Gesture 4 – Gesture 2 – Gesture 5	86.67
Gesture 1 – Gesture 3 – Gesture 2 – Gesture 4 – Gesture 5	93.33
Gesture 1 – Gesture 3 – Gesture 2 – Gesture 5 – Gesture 4	93.33
Gesture 1 – Gesture 3 – Gesture 5 – Gesture 2 – Gesture 4	100.00
Gesture 1 – Gesture 3 – Gesture 5 – Gesture 4 – Gesture 2	93.33
Gesture 1 – Gesture 4 – Gesture 3 – Gesture 5 – Gesture 2	86.67
Gesture 1 – Gesture 4 – Gesture 3 – Gesture 2 – Gesture 5	86.67
Gesture 1 – Gesture 4 – Gesture 2 – Gesture 3 – Gesture 5	100.00
Gesture 1 – Gesture 4 – Gesture 2 – Gesture 5 – Gesture 3	100.00
Gesture 1 – Gesture 4 – Gesture 5 – Gesture 2 – Gesture 3	93.33
Gesture 1 – Gesture 4 – Gesture 5 – Gesture 3 – Gesture 2	100.00
Gesture 1 – Gesture 5 – Gesture 4 – Gesture 3 – Gesture 2	86.67
Gesture 1 – Gesture 5 – Gesture 4 – Gesture 2 – Gesture 3	100.00
Gesture 1 – Gesture 5 – Gesture 2 – Gesture 4 – Gesture 3	93.33
Gesture 1 – Gesture 5 – Gesture 2 – Gesture 3 – Gesture 4	86.67
Gesture 1 – Gesture 5 – Gesture 3 – Gesture 2 – Gesture 4	100.00
Gesture 1 – Gesture 5 – Gesture 3 – Gesture 4 – Gesture 2	86.67
Gesture 2 – Gesture 1 – Gesture 3 – Gesture 2 – Gesture 4	93.33
Average	**92.27**

TH-1870_02610203

system for movement epenthesis and co-articulation detection and subsequent recognition of individual gestures in a continuous stream of gestures is an attempt that promises to perform well on different types of gesture sequences having different spatio-temporal characteristics and motion behaviour.

Table 5.7: Average gesture recognition rates for set of experiments.

Experiments and Gesture Types	**No. of training samples**	**No. of test samples**	**Recognition rates**
First (**Type I**)	450	800	89.8
Second (**Type II**)	450	800	90.9
Third (**Type III**)	250	375	94.4
Fourth (**Type III**)	250	375	90.4
Fifth (**Type IV**)	250	375	92.3

TH-1870_02610203

Chapter 6

Conclusions

In this final chapter, we present a summary of the major contributions of the work reported in the previous chapters of this thesis. Moreover, we point out a few areas that may be explored for further progress of the present research.

6.1 Summary

In the context of HCI, gesture recognition is an important research topic in the field computer vision. While glove-based gesture recognition may be considered to be a more or less solved problem, vision-based gesture recognition is still in its infancy. The thesis investigated the issues involved in the problem of vision-based hand gesture recognition and proposed methods to address some of these issues. Gesture recognition schemes were developed for recognizing gestures with different spatio-temporal characteristic for HCI applications. The schemes could identify different classes of hand gestures including static hand poses in a common vision-based platform using a single camera setup. In the proposed recognition method, local hand motions are characterized as changes in the hand shape and are represented by an FSM incorporating temporal information. On the other hand, positional displacement of the gesturing hand due to global motion is represented by a 2D gesture trajectory in space. Accordingly, gestures are recognized on the basis of FSM matching, trajectory matching, or both. The developed system is also capable of recognizing gestures in a continuous video stream via spotting of every individual gesture in

TH-1870_02610203

the sequence while discarding any unintentional movement and co-articulation occurring in between gestures.

In designing of the recognition methods throughout the thesis, we considered how a human actually behaves in reality in making a gesture as well as in recognizing it. For this, we have visually analyzed real gestures and also explored the findings in behavioral science. The main points of the work reported in this dissertation may be summed up as follows:

- In the **Introduction**, we elaborated the problems associated with dynamic hand gesture recognition. A state-of-the-art review of the existing literature on glove-based and vision-based gesture recognition was presented. The review also included methods for hand tracking in a video sequence.
- As explained in **Chapter 2**, we used the concept of object-based video abstraction for segmenting the frames into video object planes (VOPs), as used in MPEG-4. The hand is considered as a video object (VO). The proposed system exploits the inherent advantages of MPEG-4 based video object segmentation. The shape-based representation of the hand in terms of the binary alpha plane preserves finger information even in the cases when fingers are not clearly distinguishable. Based on the overall shape of the hand, we can take a recognition decision. The VOP generation method is modified from that used in MPEG-4 systems. We used AND operation rather than OR operation between horizontal and vertical candidates of a VOP so as to obtain disjoint fingers. The Hausdorff tracker used for tracking takes care of rotation and scaling of the hand thereby saving some amount of computations necessary for determining affine motions. The noteworthy feature of the recognition scheme is the use of key VOPs of a gesture sequence. The key VOPs are selected on the basis of the Hausdorff shape similarity measure. Alternatively, we proposed to use the ART shape dissimilarity measure of MPEG-7 for the same. The finite state machine modelling of the gestures, developed in this chapter, is different from the earlier FSMs since here we use only the key VOPs as states and all VOPs in between two consecutive key VOPs are included in the FSM as a temporal information for state transition. The advantage of using key VOPs is that a long gesture video sequence can be sum-

TH-1870_02610203

marized by considering only the prominent hand positions or shapes in the sequence. By doing so, we eliminate all redundant VOPs in a sequence and apply subsequent computations only on a set of a few key VOPs.

- In the proposed method for recognition of gestures having only global hand motions in **Chapter 3**, motion trajectories are extracted only from the key VOPs. Subsequently, some of the key points are selected on the basis of MPEG-7 motion trajectory approximation and representation. Key-point based trajectory representation significantly reduces memory requirement for storing the trajectory information. A feature vector comprising static and dynamic features is proposed for trajectory-guided gesture recognition. The DTW algorithm is used for mapping of key trajectory points from the template trajectory to the test and thereby features are extracted only from the key trajectory points. In the proposed method, spatio-temporal variation is collectively coped with the DTW-based nonlinear time alignment of trajectories and the MBD-based shape similarity measurement.
- In **Chapter 4**, we proposed a scheme for integrating the methods developed in Chapter 2 and Chapter 3, thus making the recognizer suitable for identifying local and/or global motion gestures in a common platform. The proposed method uses the information of displacement of centroids of key VOPs in a 2D image plane for motion interpretation. Different gesture patterns having different spatio-temporal behaviour are separated out for classification. The overall recognition is performed on the basis of following attributes.

  1. Hand shape along with temporal information.
  2. Spatial hand position in the form of gesture trajectory.
  3. Static and dynamic trajectory features.

  Fusion of these different parameters in a single system enhances recognition performance for gestures having both local and global motions. The experimental validation of the proposed system is done by recognizing some useful signs of the native Indian sign language in addition to predefined gestures having both local and global motions. The recognition rate of the proposed system on the sign language is promising.

TH-1870_02610203

- In **Chapter 5**, we proposed a method for discriminating the movement epenthesis phase from a gesture in a continuous gesture sequence. Subsequently, a movement epenthesis phase due to local hand motion is represented using the FSM-based representation. For gestures having global motions, we formulated a scheme to measure the motion behaviour in terms of fuzzy sets and devised fuzzy rules to determine whether a particular motion is a gesture phase or movement ephenthesis. Finally classification is done by the methods as proposed in Chapters 2-4.

In experimental verification of the proposed methods, gestures were selected suiting the special applications like robot control, gesture based window menu activation, etc. The selected gestures included gesture classes having different spatio-temporal characteristics. The proposed recognizer gave correct recognition with an average accuracy of more than 95%, which may be a good recognition rate particularly for HCI applications. Table 6.1 shows the performance of the proposed gesture recognizer in terms of overall accuracy rate on different types of gestures.

Table 6.1: Average Gesture Recognition Rates

Gesture types	**Average recognition rates**
Static hand postures	99.33%
Gestures with only local motion	93.30%
Gestures with only global motion	95.84%
Gestures with both local and global motion	94.30%
Static signs of native Indian sign language	95.22%
Dynamic signs of native Indian sign language	95.53%
Continuous gestures	91.56%

TH-1870_02610203

6.2 Possible Extensions

Although the presented work terminated in a working gesture recognition system for HCI, many issues arose during the work. Some of these issues require further research and are enumerated below.

1. **Illumination variation and cluttered background:** There are several constraints considered in our work and the proposed recognition methods can perform only under those constraints. One major limitation is that it totally ignores the effect of illumination change during recognition. To ease the process of tracking and segmentation, we considered relatively simple background. It will be interesting to study the performance of the proposed methods in recognizing gestures in changing illumination and cluttered background conditions. The methods might be extended to cope with these two constraints.

2. **Two handed dynamic gestures:** The proposed recognition methods can handle single-handed gestures only. It would be an interesting research problem to represent two handed gestures in FSM-based and trajectory-based representations.

3. **Depth information and occlusion:** In the proposed methods, depth information is not considered during recognition. The performance of our system may be enhanced if we consider 3D shape and motion of the hand instead of the 2D shape and motion. Another overlooked point is occlusion caused by other body parts and the hand itself. These considerations need multiple camera systems.

4. **Gesture spotting and co-articulation detection:** Gesture spotting and co-articulation detection are the main challenges in continuous dynamic gesture recognition. In the proposed methods for continuous gesture recognition, we considered the cases when a particular continuous gesture sequence is composed of component gesture sequences that have similar spatio-temporal behaviour and are free of self co-articulation. In a word, a good representation of dynamic gestures needs to be found, in which cognitive science and psychological studies may be combined in a context dependent model.

TH-1870_02610203

5. **Efficient implementation:** In the proposed methods, emphasis is given on recognition accuracy of the selected gestures rather than the efficiency of implementation. Future works may include efficient real-time implementation of the developed methods.

6. **Gesture recognition system for multimodal interface:** We also note that a hand gesture is generally accompanied with facial expression and/or movement of other body parts so as to form different meanings to the same gesture made by the hand. Therefore, integration of hand gestures with facial expression, gaze and other naturally related modes of communication in a multimodal interface is desired.

7. **Key VOP based gesture synthesis framework:** The concept of key VOPs may be extended to *gesture synthesis*, that might be used for gesture animation.

Future research may resolve all these issues to make robust and efficient hand gesture recognition systems for HCI and other applications.

TH-1870_02610203

Appendix A

A.1 Distance Transformation Algorithm

To calculate the Hausdorff distance for object matching, it is necessary to know the Euclidean distance of every pixel in the object from its nearest edge pixel. This distance for an edge pixel is obviously 0, while its horizontal and vertical neighbors, if not edge pixels themselves, have a distance of 1. For diagonal neighbors, the corresponding distance is $\sqrt{2}$ if they or their horizontal or vertical neighbors are not edge pixels.

Unfortunately, computing these distances is a global operation and computationally expensive. An algorithm that operates locally and approximates the Euclidean distance well enough is described in [Borgefors (1986), Borgefors (1988)]. This algorithm uses small masks containing integer approximations of distances in a small neighborhood thereby converting a binary image to an approximate distance image. This is called distance transformation (DT). As shown in Figure. A.1, there are two such masks, Chamfer 3-4 and Chamfer 5-7-11, that are used in the process. The horizontal, vertical and the diagonal distances in each of these masks are indicated in the figure. For example, the horizontal and vertical distances for Chamfer 3-4 are 3 each and the diagonal distance is 4. This gives a ratio of 1.333 compared to 1.414 for Euclidean (exact) distances.

The DT is initialized by assigning zero to edge pixels and infinity or a suitable large number to non-edge pixels. In two iterations, the distances are calculated by centering the mask at each pixel in turn and updating the distance of this pixel. As shown in Figure. A.2, the intensity values in the distance-transformed image represent the distances of the object pixels from their nearest edges in the original image.

TH-1870_02610203

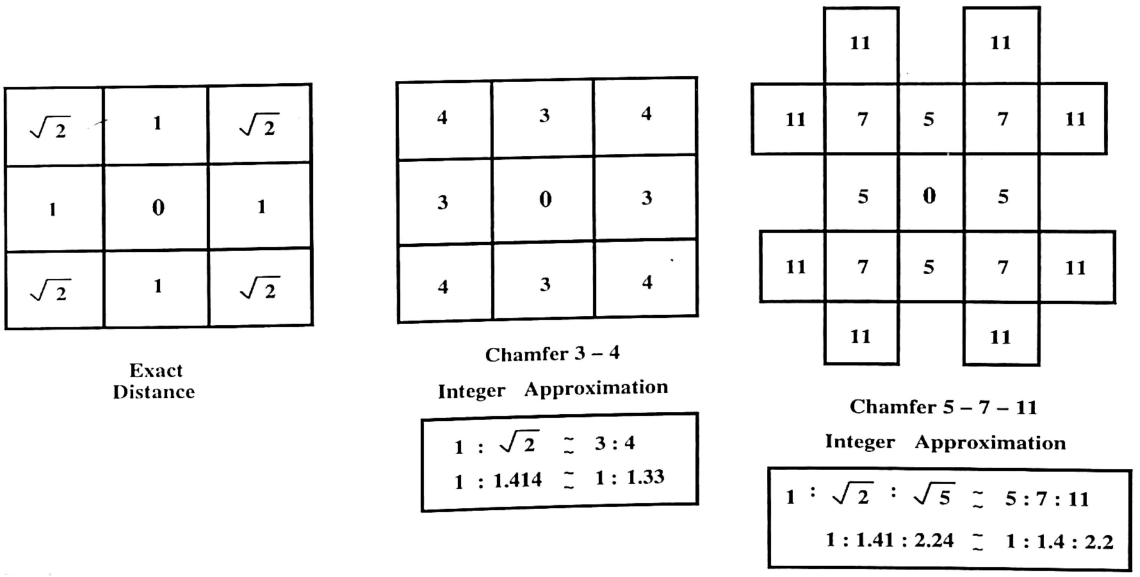


Figure A.1: Chamfer masks usedfor computing distance transformation.

Following two algorithms may be used for distance transform computation. Here mask Chamfer 3-4 is used for approximation of distances.

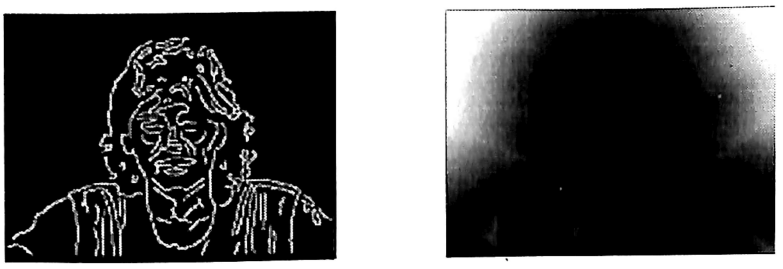
Parallel DT algorithm

If the DT is computed by parallel propagation of local distances, then at every iteration each pixel obtains a new value using the expression:

$$\begin{aligned} v_{i,j}^{k} = \min \Big(& v_{i-1,j-1}^{k-1} + 4, v_{i-1,j}^{k-1} + 3, v_{i-1,j+1}^{k-1} + 4, \\ & v_{i,j-1}^{k-1} + 3, v_{i,j}^{k-1}, v_{i,j+1}^{k-1} + 3, \\ & v_{i+1,j-1}^{k-1} + 4, v_{i+1,j}^{k-1} + 3, v_{i+1,j+1}^{k-1} + 4 \Big) \end{aligned} \tag{A.1}$$

where $v_{i,j}^{k}$ is the value of the pixel at position (i, j) in iteration k. The iterations continue until there is no more change in the value. The number of iterations is proportional to the longest distance occurring in the image.

TH-1870_02610203



Edge Image Distance Transform

Figure A.2: Edge image and corresponding distance transformed image.

Sequential DT algorithm

The sequential DT algorithm also starts from the zero/infinty image. Two passes are made over the image – first "forward" from left to right and from top to bottom; and then "backward" from right to left and from bottom to top, as described below.

Forward:

for $i = 2, \ldots\ldots$, Max_rows, DO

for $j = 2, \ldots\ldots$, Max_columns, DO

$v_{i,j} = \text{minimum}\ (v_{i-1,j-1} + 4, v_{i-1,j} + 3, v_{i-1,j+1} + 4, v_{i,j-1} + 3, v_{i,j})$.

Backward:

for $i = \text{Max_rows} - 1, \ldots\ldots, 1$ DO

for $j = \text{Max_columns} - 1, \ldots\ldots, 1$ DO

$v_{i,j} = \text{minimum}\ (v_{i,j}, v_{i,j+1} + 3, v_{i+1,j-1} + 4, v_{i+1,j} + 3, v_{i+1,j+1} + 4)$

TH-1870_02610203

Appendix B

B.1 MPEG-7 Motion Trajectory Representation

Motion trajectory describes the displacements of objects in time where an object is defined as any spatio-temporal region or set of spatio-temporal regions whose trajectory is relevant in the context in which it is used [Manjunath *et al.* (2002)]. Generally, the successive spatio-temporal positions of an object are represented by the positions of one representative point inside the object, such as its center of mass.

Description

The trajectory model is a first or second-order piecewise approximation of the spatial positions of the representative point along time, as given below. The spatial position of the object along x-dimension (*i.e.*, the coordinate along the x-axis) is approximated as

- First order approximation :

$$x(t) = x_i + v_i(t - t_i), \text{where } v_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \tag{B.1}$$

- Second order approximation :

$$x(t) = x_i + v_i(t - t_i) + \frac{1}{2}a_i(t - t_i)^2 \text{ where } v_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} - \frac{1}{2}a_i(t_{i+1} - t_i) \tag{B.2}$$

TH-1870_02610203

The y and z coordinates are also approximated using similar equations. Here v_i and a_i represent the velocity and the acceleration respectively, considered constant over $[t_i, t_{i+1}]$; (x_i, y_i) and (x_{i+1}, y_{i+1}) are object positions at times t_i and t_{i+1}.

On the basis of this model, the core of the description is a set of *key points*, representing the successive spatio-temporal positions of the concerned object, while the coordinates of the trajectory points in between these key points are interpolated using the above equations. Key points are defined by their coordinates in space and time. By default, linear interpolation is used as shown in equation B.1. When nonlinear interpolation is done between two key points in one spatial dimension, the corresponding interpolating parameter, a_i, is added to specify the second-order function of time as given in Equation B.2.

Figure B.1 illustrates the method adopted for trajectory representation. It shows on the left a modeled trajectory, and on the right the corresponding bitstream structure. In the bitstream, all the time stamps are given first. This is followed by key point positions with optional interpolating data for each spatial dimension (ord1, ord2 in the figure specify the type of interpolation used on the current interval: first or second order). The description size is roughly proportional to the number of key points and depends on the resolution of the coordinate system. Using the flexibility of the syntax, one can choose the number and temporal instances of key points to best adapt the description to the content and/or application. The total number of key points can be chosen so that the global precision and compactness required by the application is met, and variable time intervals between key points can be chosen to match the local trajectory's smoothness.

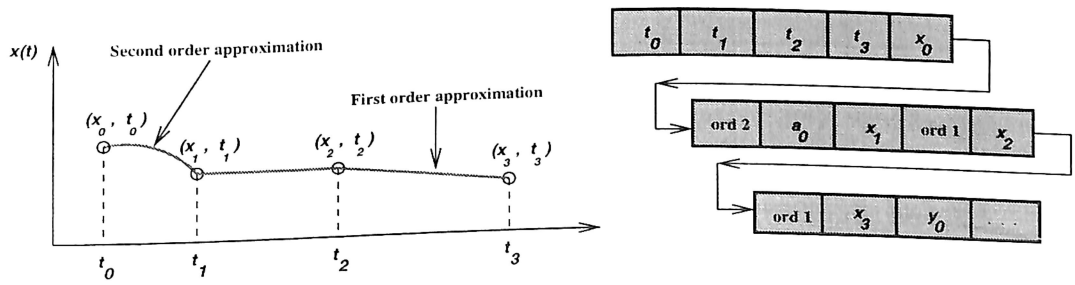


Figure B.1: Example of motion trajectory representation (One-dimensional) and corresponding bitstream structure.

TH-1870_02610203

Selection of key points

Key points are obtained by merging adjacent approximation intervals. Following two algorithms may be used for this purpose; appropriate algorithm is chosen on the basis of the constraints of the application.

- *Sequential algorithm*: The system starts with an approximation interval, which is the union of the first two intervals. It sequentially adds the following points to expand this interval, until the interpolation error exceeds a given threshold. The process repeats for every three successive points in the trajectory path.

- *Recursive Algorithm*: The system starts with an approximation interval containing all the points, and splits the interval into two at the position where the interpolation error is maximum. The splitting process is recursively repeated on each pair of new intervals. It stops either when the desired description size is reached or when the error drops below a given threshold.

B.2 Euclidean and Mahalanobis Distances

The most commonly used distance measurement is the *Euclidean distance*. The Euclidean distance between an input vector $\mathbf{x}_n$ and a prototype vector $\mathbf{y}_j$ in $\Re^d$ is given as

$$ED(\mathbf{x}_n, \mathbf{y}_j) = (\mathbf{x}_n - \mathbf{y}_j)^T(\mathbf{x}_n - \mathbf{y}_j) = \sum_{\varphi=1}^{d}(x_{n\varphi} - y_{j\varphi})^2 \tag{B.3}$$

which is simply the second Euclidean norm. A different distance measurement is the *Mahalanobis distance* which is given as

$$MD(\mathbf{x}_n, \mathbf{y}_j) = (\mathbf{x}_n - \mathbf{y}_j)^T \left[\mathbf{Cov}\right]_j^{-1} (\mathbf{x}_n - \mathbf{y}_j) \tag{B.4}$$

where $[\mathbf{Cov}]_j$ is the covariance matrix of the j^{th} cluster. This distance measurement is capable of taking into account the orientation (spread of data points) of a cluster in the feature space and is widely used in pattern recognition problems dealing with cluster analysis.

TH-1870_02610203

Publication

C.1 Research papers published/accepted related to this thesis work

1. M.K. Bhuyan, D. Ghosh and P.K. Bora, "Hand Motion Tracking and Trajectory Matching for Dynamic Hand Gesture Recognition" *Journal of Experimental and Theoretical Artificial Intelligence*, Taylor and Francis, vol. 18(4), pp. 435-447, 2006.
2. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Continuous Hand Gesture Segmentation and Co-articulation Detection", *Computer Vision, Graphics and Image Processing, Springer-Verlag*, vol.4338, pp. 564-575, 2006.
3. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Estimation of 2D Motion Trajectories from Video Object Planes and Its Application in Hand Gesture Recognition", *Lecture Notes in Computer Science, Springer-Verlag*, vol. LNCS 3776, pp. 509-514, 2005.
4. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Key Video Object Plane Selection by MPEG-7 Visual Shape Descriptor for Summarization and Recognition of Hand Gestures", *Proceedings of the Fourth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, pp. 638-643, 2004.
5. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Finite State Representation of Hand Gestures using Key Video Object Plane", *Proceedings of the IEEE Region 10 – Asia-Pacific Conference (TENCON)*, vol. A, pp. 579-582, 2004.
6. M. K. Bhuyan, D. Ghosh and P. K. Bora, "A Novel Scheme for Hand Gesture Recognition by Extracting Key Video Object Plane", *Proceedings of the International Conference on Systemics, Cybernetics and Informatics (ICSCI)*, vol. 1, pp. 130-133, 2004.

TH-1870_02610203

7. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Trajectory Guided Recognition of Hand Gestures for Human Computer Interface", *Proceedings of Second Indian International Conference on Artificial Intelligence (IICAI)*, pp. 312-327, 2005.

8. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Co-articulation Detection in Hand Gestures", *Proceedings of the IEEE Region 10 – Asia-Pacific Conference (TENCON)*, pp. 2328- 2333, 2005.

9. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Designing of Human Computer Interactive Platform for Robotic Applications", *Proceedings of the IEEE Region 10 – Asia-Pacific Conference (TENCON)*, pp. 682-687, 2005.

10. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Automatic Video Object Plane Generation for Recognition of Hand Gestures", *Proceedings of the International Conference on Systemics, Cybernetics and Informatics (ICSCI)*, pp. 147-152, 2005.

11. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Threshold Finite State Machine for Vision Based Gesture Recognition", *Proceedings of the IEEE Indicon Conference*, pp. 379-382, 2005.

12. M.K. Bhuyan, D. Ghosh and P.K. Bora, "Feature Extraction from 2D Gesture Trajectory in Dynamic Hand Gesture Recognition", *Proceedings of the IEEE International Conferences on Cybernetics and Intelligent Systems (CIS) and Robotics, Automation and Mechatronics (RAM)*, pp. 717-722, 2006.

13. M. K. Bhuyan, D. Ghosh and P. K. Bora, "State Selection via DTW for Finite State based Gesture Recognition", *Proceedings of the 12th National Conference on Communications (NCC)*, pp. 1-4, 2006.

14. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Recognition of Wide Classes of Hand Gestures for Human Computer Interaction", *Proceedings of International Conferences on Man Machine Systems (ICoMMS)*, Langkawi, Malaysia, 15-16 September, 2006.

15. M. K. Bhuyan, D. Ghosh and P. K. Bora, "A Framework for Hand Gesture Recognition with Applications to Sign Language", *Proceedings of IEEE Indicon 2006 Conference*, New Delhi, India, 15-17 September, 2006.

TH-1870_02610203

C.2 Research papers communicated

1. M. K. Bhuyan, D. Ghosh and P. K. Bora, "FSM based Recognition of Dynamic Hand Gestures via Gesture Summarization using Key Video Object Planes", communicated to the *Journal of Visual Language and Computing*, Date of Communication : 24.06.2005.
2. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Dynamic Hand Gesture Recognition using 2D Motion Trajectories Extracted from Video Object Planes", communicated to the *Digital Signal Processing*, Date of Communication : 04.10.2005.
3. M. K. Bhuyan, D. Ghosh and P. K. Bora, "An Integrated Scheme for Recognition of Wide Classes of Hand Gestures in a Vision Based Setup", to be communicated to IEEE Transactions on Systems, Man, and Cybernetics - Part A.
4. M. K. Bhuyan, D. Ghosh and P. K. Bora, "Recognition of Continuous Hand Gestures having Local Motions only", communicated to International conference on Computing: Theory and Applications, 2007.

TH-1870_02610203

Bibliography

AGGARWAL, J. & CAI, Q. (1997). Human motion analysis: A review. *Proceedings of the Nonrigid and Articulated Motion Workshop*, 90–102.

AHMAD, S. (1994). A usable real-time 3D hand tracker. *Proceedings of the IEEE Asilomar Conference*.

AZARBAYEJANI, A., WREN, C. & PENTLAND, A. (1996). Real-time 3D tracking of the human body. *Proceedings of the IMAGE'COM*, Bordeaux, France.

BAUDEL, T. & BEAUDOUIN-LAFON, M. (1993). Charade: Remote control of objects using free-hand gestures. *Communication of the ACM*, **36 (7)**, 28–35.

BERRY, G. (1998). *Small-wall: A Multimodal Human Computer Intelligent Interaction Test Bed with Applications*. Master's thesis, Department of ECE, University of Illinois at Urbana-Champaign.

BEZDEK, J.V. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.

BIRK, H. & MOESLUND, T.B. (1996). *Recognizing Gestures from the Hand Alphabet using Principal Component Analysis*. Master's thesis, Laboratory of Image Analysis, Aalborg University, Denmark.

BLACK, M.J. & JEPSON, A. (1998a). Recognition temporal trajectories using condensation algorithm. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 16–21.

TH-1870_02610203

BLACK, M.J. & JEPSON, A.D. (1998b). A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. *Proceedings of the European Conference on Computer Vision*, **1**, 909–924.

BOBER, M. (2001). MPEG-7 visual shape descriptors. *IEEE Transaction on Circuits and Systems for Video Technology*, **11(6)**, 716–719.

BOBICK, A.F. & DAVIS, J. (1996). Real-time recognition of activity using temporal templates. *Proceedings of the IEEE CS Workshop on Applications of Computer Vision*, 39–42.

BOBICK, A.F. & WILSON, A.D. (1997). A state-based approach to the representation and recognition of gesture. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **19 (12)**, 1325–1337.

BOEHM, K., BROLL, W. & SOKOLOWICZ, M. (1994). Dynamic gesture recognition using neural networks. *Proceedings of the SPIE Conference on Electronic Imaging Science and Technology*.

BORECZKY, J.S. & LAWRENCE, A.R. (1996). Comparision of video shot boundary detection techniques. *Proceedings of the SPIE.*, **2670**, 170–179.

BORGEFORS, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, **34**, 344–371.

BORGEFORS, G. (1988). Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **10 (6)**, 849–857.

BRAND, M., OLIVER, N. & PENTLAND, A. (1997). Coupled hidden Markov models for complex action recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 994–999.

BREGLER, C. (1997). Learning and recognizing human dynamics in video sequences. *Proceedings of the IEEE international Conference on Computer Vision and Pattern Recognition*.

TH-1870_02610203

CANNY, J.F. (1986). A comutational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8(6)**, 679–698.

CHANDRAN, S. & SAWA, A. (2004). Real time detection and understanding of isolated protruded fingers. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 152 – 157.

CHEN, F.S., FU, C.M. & HUANG, C.L. (2003). Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and Vision Computing*, **21**, 745–758.

CHOUDHURY, T., CLARKSON, B., JEBARA, T. & PENTLAND, A. (1999). Multimodal person recognition using unconstrained audio and video. *Proceedings of the Second Conference on Audio and Video based Biometric Person Authentication*, 1–6.

CIPOLLA, R., OKAMOTO, Y. & KUNO, Y. (1993). Robust structure from motion using motion parallax. *Proceedings of the International Conference on Computer Vision*, 374–382.

CLERCUE, E., GOLDBERG, M., MADRANE, N. & MERIALDO, B. (1995). Automatic face and gestural recognition for video indexing. *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, 110–115.

COHEN, C., CONWAY, L. & KODITSCHEK, D. (1996). Dynamical system representation, generation, and recognition of basic oscillatory motion gestures. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*.

CROWLEY, J.L., BERARD, F. & COUTAZ, J. (1995). Finger tacking as an input device for augmented reality. *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 195–200.

CUI, Y. & WENG, J.J. (1997). Hand segmentation using learning-based prediction and verification for hand sign recognition. *Proceedings of the IEEE CS Conference on Computer Vision and Pattern Recognition*, 88–93.

TH-1870_02610203

DARRELL, T. & PENTLAND, A. (1994). Recognition of space-time gestures using a distributed representation. *M.I.T. Media Laboratory Vision and Modeling Group*, **Technical Report No. 197**.

DARRELL, T. & PENTLAND, A. (1995). Attention-driven expression and gesture analysis in an interactive environment. *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 135–140.

DARRELL, T. & PENTLAND, A. (1996). Active gesture recognition using partially observable Markov decision processes. *Proceedings of the IEEE International Conference on Pattern Recognition*, 984–988.

DARRELL, T., ESSA, I. & PENTLAND, A. (1995). Task-specific gesture analysis in real-time using interpolated views. *M.I.T Media Laboratory Perceptual Computing Group*, **Technical Report No. 364**.

DARRELL, T., ESSA, I. & PENTLAND, A. (1996). Task-specific gesture analysis in real-time using interpolated views. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **18(12)**, 1236–1242.

DAVIS, J. & SHAH, M. (1993). Gesture recognition. *University of Central Florida, Orlando*, **Technical Report No. CS-TR-93-11**.

DAVIS, J. & SHAH, M. (1994a). Recognizing hand gestures. *Proceedings of the European Conference on Computer Vision (ECCV)*, 331–340.

DAVIS, J. & SHAH, M. (1994b). Visual gesture recognition. *Vision, Image, and Signal Processing*, **141(2)**, 101–106.

DONATO, G.A., BARTLETT, M.S., HAGER, J.C., EKMAN, P. & SEJNOWSKI, T.J. (1999). Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21(10)**, 974–989.

DOWNTON, A.C. & DROUET, H. (1991). Image analysis for model-based sign language coding. *Progress in Image Analysis and Processing II: Proceedings of the Sixth International Conference on Image Analysis and Processing*, 637–644.

TH-1870_02610203

DUDA, R.O., HART, P.E. & STORK, D.G. (2002). *Pattern Classification*. John Wiley and Sons, Inc., New York, 2nd edn.

EFRON, D. (1941). *Gesture and Environments*. King Crown Press.

EISENSTEIN, J. & DAVIS, R. (2005). Gestural cues for sentence segmentation. *Technical report, MIT AI Memo*, 1–14.

EROL, B. & KOSSENTINI, F. (2000). Automatic key video object plane selection using the shape information in the MPEG-4 compressed domain. *IEEE Transaction on Multimedia*, **2(2)**, 129–138.

EROL, B. & KOSSENTINI, F. (2001). Local motion descriptors. *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, 467–472.

ETOH, M., TOMONO, A. & KISHINO, F. (1991). Stereo-based description by generalized cylinder complexes from occluding contours. *Systems and Computers in Japan*, **22(12)**, 79–89.

FAHN, C.S. & SUN, H. (2005). Development of a data glove with reducing sensors based on magnetic induction. *IEEE Transactions on Industrial Electronics*, **52(2)**, 585–594.

FELS, S.S. & HINTON, G.E. (1993). Glove-talk: A neural network interface between a DataGlove and a speech synthesizer. *IEEE Transactions on Neural Networks*, **4(1)**, 2–8.

FELS, S.S. & HINTON, G.E. (1997). Glove-talk II: A neural network interface which maps gestures to parallel format speech synthesizer controls. *IEEE Transactions on Neural Networks*, **9(11)**, 205–212.

FORNER-CORDERO, A., KOOPMAN, H. & DER HELM, F.V. (2006). Describing gait as a sequence of states. *Journal of Biomechanics*, **39(5)**, 948–957.

FREEMAN, W.T. & ROTH, M. (1995). Orientation histograms for hand gesture recognition. *Proceedings of the IEEE Workshop on Automatic Face and Gesture Recognition*.

FREEMAN, W.T. & WEISSMAN, C.D. (1995). Television control by hand gestures. *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*.

TH-1870_02610203

GAVRILA, D.M. (1999). The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, **73(1)**, 82–98.

GAVRILA, D.M. & DAVIS, L.S. (1995). Towards 3D model-based tracking and recognition of human movement: A multi-view approach. *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 272–277.

GEORGILA, K., SGARBAS, K., TSOPANOGLOU, A., FAKOTAKIS, N. & KOKKINAKIS, G. (2003). A speech-based human-computer interaction system for automating directory assistance services. *International Journal of Speech Technology*, **6**, 145 – 159.

GIBET, S., COURTY, N. & KAMP, J. (2005). Gesture in human-computer interaction and simulation. *Springer Lecture Notes in Artificial Intelligence*, **LNAI 3881**, 344.

HEAP, T. & HOGG, D. (1996). Towards 3D hand tracking using a deformable model. *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 140–145.

HIRCHE, S. & BUSS, M. (2004). Telepresence control in packet switched communication networks. *Proceedings of the IEEE International Conference on Control Applications*, **1**, 236–241.

HOLDEN, E.J., LEE, G. & OWENS, R. (2005). Australian sign language recognition. *Machine Vision and Applications*, **16(5)**, 312–320.

HONG, P., TURK, M. & HUANG, T.S. (2000). Gesture modelling and recognition using finite state machines. *Proceeding of the IEEE Conference on Face and Gesture Recognition*, 410–415.

HU, W., TAN, T., WANG, L. & MAYBANK, S. (2004). A survey on visual surveillance of object motion and behaviours. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, **34(3)**, 334–352.

HUANG, X. & WENG, J. (2004). Motivational system for human-robot interaction. *Springer-Verlag*, **LNCS 3058**, 17–27.

TH-1870_02610203

HUTTENLOCHER, D.P., KLANDERMAN, G.A. & RUCKLIDGE, W.J. (1993a). Comparing images using the Hausdorff distance. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **15 (9)**, 850–863.

HUTTENLOCHER, D.P., NOH, J.J. & RUCKLIDGE, W.J. (1993b). Tracking non-rigid objects in complex scene. *Proceedings of the fourth International Conference of Computer Vision*, 93–101.

ISARD, M. & BLAKE, A. (1996). Contour tracking by stochastic propagation of conditional density. *Proceedings of the European Conference on Computer Vision*, 343–356.

JAIMES, A. & SEBE, N. (2005). Multimodal human computer interaction: A survey. *Springer-Verlag*, **LNCS 3766**, 1–15.

JAIN, A.K. (1989). *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ.

JERDE, T.E., SOECHTING, J.F. & FLANDERS, M. (2003). Coarticulation in fluent fingerspelling. *The Journal of Neuroscience*, **23 (6)**, 2383 – 2393.

KAHOL, K., TRIPATHI, P. & PANCHANATHAN, S. (2004). Automated gesture segmentation from dance sequences. *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, 883–888.

KAMARAINEN, J.K., KYRKI, V. & KALVIAINEN, H. (2006). Invariance properties of gabor filter-based features-overview and applications. *IEEE Transactions on Image Processing*, **15(5)**, 1088–1099.

KANEKO, T. & HORI, O. (2003). Feature selection for reliable tracking using template matching. *Proceedings of the Computer Vision and Pattern Recognition*, **1**, 796–802.

KANG, H., LEE, C. & JUNG, K. (2004). Recognition-based gesture spotting in video games. *Pattern Recognition Letters*, **25**, 1701–1714.

TH-1870_02610203

KATKERE, A., HUNTER, E., KURAMURA, D., SCHLENZIG, J., MOEZZI, S. & JAIN, R. (1994). Robogest: Telepresence using hand gestures. *University of California, San Diego, Visual Computing Laboratory*, **Technical Report No. VCL-94-104**.

KATO, M., CHEN, Y.W. & XU, G. (2006). Articulated hand tracking by PCA-ICA approach. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 329–334.

KENDON, A. (1988). How gestures can become like words. Cross cultural perspectives in nonverbal communication. *F. Potyados*, **20 (12)**, 131–141.

KENDON, A. (1990). *Conducting Interaction*. Cambridge University Press.

KEVIN, N.Y.Y., RANGANATH, S. & GHOSH, D. (2004). Trajectory modeling in gesture recognition using cybergloves and magnetic trackers. *Proceedings of the IEEE Region 10 – Asia-Pacific Conf (TENCON)*, **A**, 571–574.

KIM, C. & HWANG, J.N. (2002). Object-based video abstraction for video surveillance systems. *IEEE Transaction on Circuits and Systems for Video Technology*, **12(12)**, 1128–1138.

KIM, I.C. & CHIEN, S.I. (2001). Analysis of 3D hand trajectory gestures using stroke-based composite hidden Markov models. *Applied Intelligence*, **15**, 131–143.

KJELDESEN, R. & KENDER, J. (1996). Toward the use of gesture in traditional user interfaces. *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, 151–156.

KJELDSEN, R. & KENDER, J. (1996). Finding skin in color images. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 312–317.

KNAPP, M.L. & HALL, J.A. (1997). *Nonverbal Communication in Human Interaction*. Harcourt Brace Publishers, Ft. Worth, Texas, 4th edn.

KOCH, R. (1993). Dynamic 3D scene analysis through synthetic feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15(6)**, 556–568.

TH-1870_02610203

KONG, W.W. & RANGANATH, S. (2004). 3D hand trajectory recognition for signing exact english. *Proceedings of the sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 535–540.

KRUEGER, M. (1991). *Artificial Reality II*. Addison-Wesley.

KRUEGER, M. (1993). Environmental technology: Making the real world virtual. *Communications of the ACM*, **36(7)**, 36–37.

KUCH, J.J. & HUANG, T.S. (1995). Vision-based hand modeling and tracking for virtual teleconferencing and telecollaboration. *Proceedings of the IEEE International Conference on Computer Vision*, 666–671.

LANDAY, J. & MYERS, B. (1998). Applications of face and gesture recognition for human-computer interaction. *Proceedings of the Sixth ACM International Multimedia Conference on Face/Gesture Recognition and their Applications*, 20–27.

LEE, C. & XU, Y. (1996). *Online, interactive learning of gestures for human robot interface*. Carnegie Mellon University, The Robotics Institute, Pittsburgh, Pennsylvania, USA.

LEE, H.K. & KIM, J.H. (1999). An HMM- based threshold model approach for gesture recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **21(10)**, 961–973.

LEE, J. & KUNII, T.L. (1995). Model-based analysis of hand posture. *IEEE Computer Graphics and Applications*, 77–86.

LIANG, R.H. & OUHYOUNG, M. (1998). Real-time continuous gesture recognition system for sign language. *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 558–567.

LIDDELL, S. & JOHNSON, R. (1989). American sign language: The phonological base. *Sign Language Studies*, **64**, 195–277.

TH-1870_02610203

LUPATINI, G., SARACENO, C. & LEONARDI, R. (1998). Scene break detection: a comarision. *Proceedings of the RIDE*.

MAES, P., DARRELL, T., BLUMBERG, B. & PENTLAND, A. (1997). The ALIVE system: Wireless, full-body interaction with autonomous agents. *Multimedia Systems*, **5**, 105–112.

MAGGIONI, C. (1993). A novel gestural input device for virtual reality. *IEEE Annual Virtual Reality International Symposium*, 118–124.

MAGGIONI, C. (1995). Gesture Computer - New ways of operating a computer. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 166–171.

MALKAWI, A.M. & SRINIVASAN, R. (2005). A new paradigm for human-building interaction: the use of CFD and augmented reality. *Journal of Automation in Construction*, **14**, 71–84.

MANJUNATH, B.S., SALEMBIER, P. & SIKORA, T., eds. (2002). *Intoduction to MPEG-7, Multimedia Content Description Interface*. John Wiley and Sons Ltd., England.

MCKENNA, S.J. & MORRISON, K. (2004). A comparison of skin history and trajectory-based representation schemes for the recognition of user-specified gestures. *Pattern Recognition*, **37**, 999 – 1009.

MCNEILL, D. (1992). *Hand and Mind*. University of Chicago Press.

MEIER, T. & NAGAN, K.N. (1998). Automatic segmentation of moving objects for video object plane generation. *IEEE Transaction on Circuits and Systems for Video Technology*, **8 (5)**, 525–538.

MEIER, U., STIEFELHAGEN, R., YANG, J. & A.WAIBEL (2000). Toward unrestricted lip reading. *International Journal of Pattern Recognition and Artificial Intelligence*, **14(5)**, 571–585.

TH-1870_02610203

MINERS, B.W., BASIR, O.A. & KAMEL, M.S. (2005). Understanding hand gestures using approximate graph matching. *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, **35 (2)**, 239 – 248.

MORGUET, P. & LANG, M. (1998). Spotting dynamic hand gestures in video image sequences using hidden Markov models. *Proceedings of the International Conference on Image Processing (ICIP)*, 193–197.

NAM, Y. & WOHN, K. (1996). Recognition of space-time hand-gestures using hidden Markov model. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 51–58.

NAM, Y. & WOHN, K. (1997). Recognition of hand gestures with 3D, nonlinear arm movements. *Pattern Recognition Letters*, **18(1)**, 105–113.

NISHIMURA, T. & OKA, R. (1996). Towards the integration of spontaneous speech and gesture based on spotting method. *Proceedings of the IEEE/SICE/RSJ International Conference on Multisensor Fusion Integration Intelligent System*, 433–437.

NOLKER, C. & RITTER, H. (1998). Illumination independent recognition of deictic arm postures. *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society,*, 2006–2011.

OUDOT, L., PREVOST, L. & MOISES, A. (2004). Self-supervised adaptation for on-line text recognition. *Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, 9–13.

PAVLOVIC, V.I., SHARMA, R. & HUANG, T.S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **19 (7)**, 677–695.

PEDERSEN, E.E. (1991). Space in Danish sign language - the semantics and morphosyntax of the use of space in a visual language. *International Studies on Sign Language Research and Communication of the Deaf*, **19**.

TH-1870_02610203

PENTLAND, A. (2000). Looking at people. *Communication of the ACM*, **43(3)**, 35–44.

PENTLAND, A. & LIU, A. (1999). Modeling and prediction of human behaviour. *Neural Computation*, **11(1)**, 229–242.

POLANA, R. & NELSON, R. (1994). Low level recognition of human motion. *Proceeding of the IEEE CS Workshop on Motion of Non-Rigid and Articulated Objects*, 77–82.

QUAM, D.L. (1990). Gesture recognition with a data glove. *Proceedings of the IEEE Conference on National Aerospace and Electronics*, **2**, 755–760.

QUEK, F. (1994). Toward a vision-based human gesture interface. *Proceedings of the International Conference on Virtual Reality Software Technolgy*, 17–31.

QUEK, F., MCNEILL, D., BRYLL, R., KIRBAS, C., ARSLAN, H., MCCULLOUGH, K.E., FURUYAMA, N. & ANSARI, R. (2000). Gesture, speech, and gaze cues for discourse segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **2**, 247–254.

QUEK, F.K.H. (1995). Eyes in the interface. *Image and Vision Computing*, **13**.

QUEK, F.K.H., MYSLIWIEC, T. & ZHAO, M. (1995). Finger mouse: A freehand pointing interface. *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 372–377.

RABINER, L.R. & JUANG, B. (1993). *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, N.J.

RAMAMOORTHY, A., VASWANI, N., CHAUDHURY, S. & BANERJEE, S. (2003). Recognition of dynamic hand gestures. *Pattern Recognition*, **36**, 2069–2081.

REHG, J. & KANADE, T. (1995). Model-based tracking of self-occluding articulated objects. *Proceedings of the IEEE International Conference on Computer Vision*, 612–617.

REHG, J.M. & KANADE, T. (1993). Digiteyes: Vision-based human hand tracking. *School of Computer Science, Carnegie Mellon University*, **Technical Report CMU-CS-93-220**.

TH-1870_02610203

REHG, J.M. & KANADE, T. (1994). Visual tracking of high DOF articulated structures: An application to human hand tracking. *Proceedings of the Third European Conference on Computer Vision*, 35–46.

RITTSCHER, J. & BLAKE, A. (1999). Classification of human body motion. *IEEE International Conference on Computer Vision.*

ROSALES, R. & SCLAROFF, S. (2000). Inferring body pose without tracking body parts. *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, **II**, 721–727.

ROSE, R.C. (1992). Discriminant wordspotting techniques for rejection non-vocabulary utterances in unconstrained speech. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, **II**, 105–108.

ROSE, R.C. (1996). Word spotting from continuous speech utterances. *Automatic Speech and Speaker Recognition - Advanced Topics*, 303–330.

ROURKE, O. & BADLER, N.L. (1980). Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2**, 522–536.

SAKOE, H. & CHIBA, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transaction on Acoustics, Speech and Signal Processing*, **26 (1)**, 43–49.

SCHLENZIG, J., HUNTER, E. & JAIN, R. (1995). Recursive spatio-temporal analysis: Understanding gestures. *Visual Computing Laboratory, University of San Diego, California*, **Technical Report No. VCL-95-109**.

SEBE, N., LEW, M.S. & HUANG, T.S. (2004). The state of the art in human-computer interaction. *Springer-Verlag*, **LNCS 3058**, 1–6.

SEGEN, J. & KUMAR, S. (1998). GestureVR: Vision-based 3D hand interface for spatial interaction. *Proceedings of the Sixth ACM International Conference on Multimedia*, 455–464.

TH-1870_02610203

SHAMAIE, A. & SUTHERLAND, A. (2001). Classification of co-articulated and partly occluded hand gestures. *Proceedings of the Irish Machine Vision and Image Processing Conference*.

SHAMAIE, A., HAI, W. & SUTHERLAND, A. (2001). Hand gesture recognition for HCI. *ERCIM News (on line edition), http://www.ercim.org/publication/Ercim_News*, **46**.

SHRIBERG, E., STOLCKE, A., HAKKANI-TUR, D. & TUR, G. (2000). Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, **32**.

STARNER, T. & PENTLAND., A. (1995). Real-time American sign language recognition from video using hidden Markov models. *Technical Report, M.I.T Media Laboratory Perceptual Computing Section*, **Technical Report No. 375**.

STARNER, T., WEAVER, J. & PENTLAND, A. (1998). Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **20(12)**, 1371–1375.

STOKOE, W. (1960). *Sign Language Structure*. University of Buffalo Press.

STOLL, P. & OHYA, J. (1995). Applications of HMM modeling to recognizing human gestures in image sequences for a man-machine interface. *Proceedings of the IEEE International Workshop on Robot and Human Communication*.

STURMAN, D.J. & ZELTZER, D. (1994). A survey of glove-based input. *IEEE Computer Graphics and Applications*, **14**, 30–39.

TAKAHASHI, K., SEKI, S. & OKA, R. (1992). Spotting recognition of human gestures from motion images. *The Insttitute of Electronics, Information, and Communication Engineers*, **Technical Report IE92-134**, 9–16.

TAKAHASHI, T. & KISHINO, F. (1991). Hand gesture coding based on experiments using a hand gesture interface device. *In SIGCHI Bul*, **23**, 67–73.

THALMANN, N.M. & THALMAN, D. (1990). *Computer Animation: Theory and Practice*. Springer-Verlag, New York, 2nd edn.

TH-1870_02610203

THOMPSON, D. (1981). Biomechanics of the hand. *Perspectives in Computing*, **1**, 12–19.

THORISSON, K. (1994). Face-to-face communication with computer agents. *Proceedings of the AAAI Spring Symposium on Believable Agents*, 86–90.

TRIESCH, J. & MALSBURG, C. (1996). Robust classification of hand postures against complex background. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 170–175.

TSAIG, Y. & AVERBUCH, A. (2002). Automatic segmentation of moving objects in video sequences: A region labeling approach. *IEEE Transactions on Circuits and Systems for Video Technology*, **12 (7)**, 597–612.

TURK, M. (2004). Computer vision in the interface. *Communications of the ACM*, **47 (1)**, 60–67.

TURK, M. & KOLSCH, M. (2004). *Perceptual Interfaces*. Prentice Hall.

VOGLER, C. & METAXAS, D. (1998). ASL recognition based on a coupling between HMM and 3D motion analysis. *Proceedings of the International Conference on Computer Vision*, 363–369.

VOGLER, C. & METAXAS, D. (1999). Toward scalability in ASL recognition: Breaking down signs into phonemes. *Proceedings of the IEEE Gesture Workshop*.

VOGLER, C. & MEXTAXAS, D. (1997). Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 156–161.

VOGLER, C. & MEXTAXAS, D. (1999). Parallel hidden Markov models for American sign language recognition. *Proceedings of the International Conference on Computer Vision*, 1–7.

WANG, T., SHUM, H., XU, Y. & ZHENG, N. (2001). Unsupervised analysis of human gestures. *Proceedings of the IEEE Pacific Rim Conference on Multimedia*, 174–181.

TH-1870_02610203

WATSON, R. (1993). Gesture recognition techniques. *Trinity College, Department of Computer Science, Dublin*, **Technical Report No. TCD-CS-93-11**.

WILSON, A. & BOBICK, A. (1995a). Learning visual behaviour for gesture analysis. *Proceedings of the IEEE Symposium on Computer Vision*, 19–21.

WILSON, A. & BOBICK, A. (1995b). Using configuration states for the representation and recognition of gesture. *M.I.T. Media Laboratory Perceptual Computing Section*, **Technical Report No. 308**.

WILSON, A. & BOBICK, A. (1998). Recognition and interpretation of parametric gesture. *Proceedings of the IEEE International Conference on Computer Vision*, **3**, 329–336.

WREN, C., AZARBAYEJANI, A., DARRELL, T. & PENTLAND, A. (1996). Pfinder: Real-time tracking of the human body. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 51–56.

WU, Y. & HUANG, T.S. (1999a). Capturing articulated human hand motion: A divide-and-conquer approach. *Proceedings of the IEEE International Conference on Computer Vision*, 606–611.

WU, Y. & HUANG, T.S. (1999b). Vision-based gesture recognition: A review. *Lecture Notes in Computer Science, Gesture Workshop*, 1–12.

WU, Y. & HUANG, T.S. (2000a). Self-supervised learning for visual tracking and recognition of human hand. *Proceedings of the National Conference on Artificial Intelligence, (AAAI)*, 243–248.

WU, Y. & HUANG, T.S. (2000b). View-independent recognition of hand postures. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **II**, 88–94.

WU, Y. & HUANG, T.S. (2001). Hand modelling, analysis, and recognition for vision-based human computer interaction. *IEEE Signal Processing Magazine*, 51–60.

TH-1870_02610203

WU, Y., LIN, J. & HUANG, T.S. (2005). Analyzing and capturing articulated hand motion in image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27 (12)**, 1910 – 1921.

YAMATO, J., OHYA, J. & ISHII, K. (1992). Recognizing human action in time-sequential images using hidden Markov model. *Proceeding of the IEEE Conference on Computer Vision and Pattern recognition*, 379–385.

YANG, M. & AHUJA, N. (1998). Extraction and classification of visual motion pattern recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 892–897.

YANG, M.H., AHUJA, N. & TABB, M. (2002). Extraction of 2D motion trajectories and its application to hand gesture recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **24 (8)**, 1061–1074.

ZADEH, L. (1988). Fuzzy logic. *Computer*, **1(4)**, 83–93.

ZELLER., M., PHILLIPS, J.C., DALKE, A., HUMPHREY, W., SCHULTEN, K., SHARMA, R., HUANG, T.S., PAVLOVIC, V.I., ZHAO, Y., LO, Z. & CHU, S. (1997). A visual computing environment for very large scale biomolecular modeling. *Proceedings of the IEEE Conference on Application-specific Systems, Architectures and Processors (ASAP)*, 3–12.

ZHAO, L. (2001). *Synthesis and Acquisition of Laban Movement Analysis Qualitative Parameters for Communicative Gestures*. Ph.D. thesis, CIS, University of Pennsylvania.

ZIEREN, J., UNGER, N. & AKYOL, S. (2002). Hands tracking from frontal view for vision-based gesture recognition. *Proceedings of the DAGM Symposium*, 531–539.

ZIMMERMAN, T.G. & LANIER, J. (1987). A hand gesture interface device. *In ACM SIGCHI/GI*, 189–192.

TH-1870_02610203

ANSWERS OF THE REVIEWER - 1

- **Question 1:** How robust is your Hausdorff distance based tracking and key video plane detection scheme? Can you show some results against textured background?

Answer: To ease the process of tracking and segmentation, we considered relatively simple but not totally plain background. The Hausdroff tracker is computationally efficient and can track objects under cluttered background [1]. The method has been reported to be robust to noise and changes in shape of the object. As depicted in Figure 2.8 of the thesis, background filtering is done before tracking. Background filtering is one desired step when VOPs are extracted from cluttered background. Removal of background edges reduces the number of points to be considered during the process of shape change detection and thus speeds up the process of model updation and tracking. The algorithm is capable of functioning even without background removal as long as the background does not change significantly [1]. Moreover, as proposed in [2], video object planes can be generated even if the background is moving. In this particular case, it is required to develop a global background motion compensation model to generate video object planes. In this, the hierarchal block matching techniques have been applied. In our system, we do not consider this case.

Typically, key VOPs are selected such that they reflect significant changes in the shape, colour, and texture contents of a video object. Using shape content of a video object for key VOP selection has many advantages over using colour and/or texture content. First, the texture and colour of a video object are generally consistent during the object's lifespan. On the other hand, the shape of a video object may vary significantly due to the object's movement, structure, occlusion etc. Therefore, a significant change in the content of a video object is more likely to be detected if the object's shape is employed as a measure. In order to detect significant changes in the shape content, a shape similarity measure is required. For this, the Hamming or the Hausdorff distance measures may be used to estimate the difference between two hand shapes, as used in [3]. The Hamming distance measures the point-by-point difference between two shapes, whereas the Hausdorff distance measures the largest distance between the contours of two shapes. Both of these distance measures are commonly used in shape matching, and both have different implementation and computational complexities. However, unlike the Hamming distance where all the points in the objects contribute to the measure of the distance between two video objects contained in their respective VOPs, the Hausdorff distance depends only on the two object points (one in each of the two VOPs under consideration) that have the largest distance between them. Moreover, unlike the threshold used in the Hamming distance based algorithm, the threshold used in the Haus-

TH-1870_02610203

dorff distance measure for shape similarity measurement does not depend on the activity level of the video object [3]. In view of the advantages offered by the Hausdorff distance based shape similarity measure over that based on the Hamming distance, we propose to use the Hausdorff shape similarity measure for detecting key hand poses in the gesture video sequence.

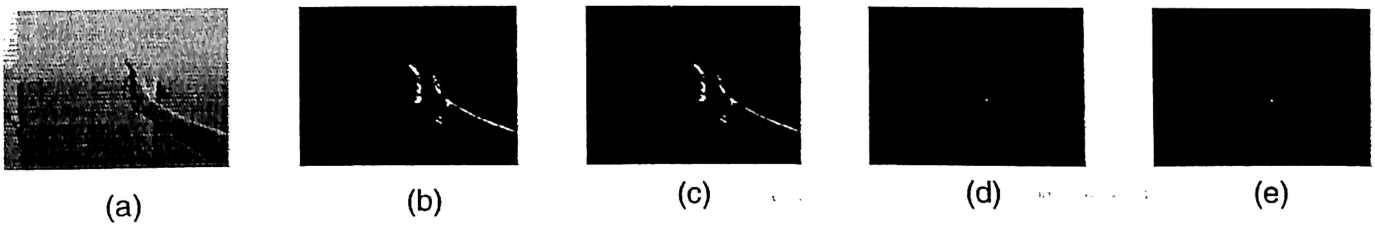


Fig. 1. Initial model generation and background filtering, (a) First frame (b) Thresholded difference image (c) Median filtering of threshold difference image (d) Thinning (e) Connected component labelling.

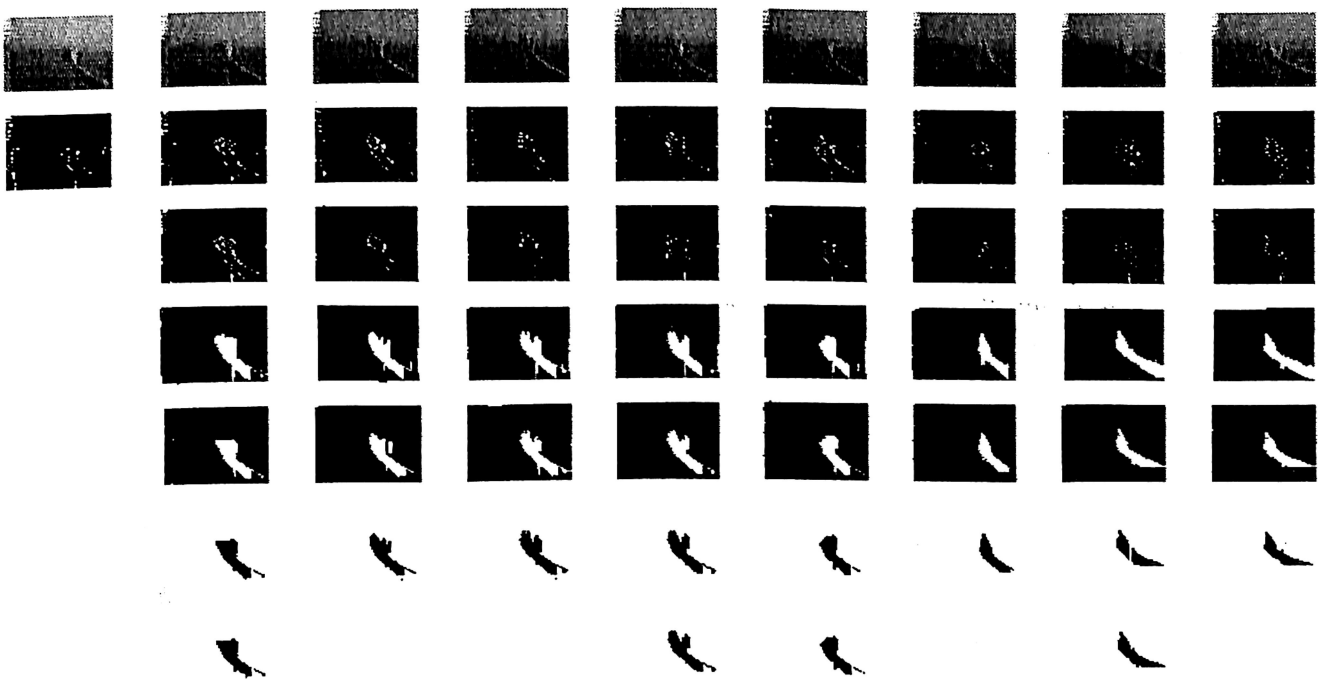


Fig. 2. Key VOP selection for a gesture video sequence under textured background.

As explained in Chapter 2 of the thesis, an initial model is necessary for tracking of video object in successive video frames of the input video sequence. An example of initial model generation in a textured background

TH-1870_02610203

is illustrated in Fig. 1. The results of key VOP selection for this image quence, is shown in Fig. 2. The first row of the figure shows the origi image sequence, second row shows the difference edge images, third row shows moving edges, fourth rows show vertical candidates of a particular VOP respectively. The AND combination of vertical and horizontal candidates of each VOP is shown in the fifth row. After morphological filtering we get the binary alpha plane corresponding to each VOP, as shown in the sixth row. The last row of each figure shows the finally extracted key VOPs with all in-between redundant frames discarded.

- **Question 2:** When you use DTW based approach, does performance of the system degrade as the number of gestures in knowledge base increases?

  Answer: Dynamic time warping (DTW) is a method for computing a non-linear time normalization between a template vector sequence and a test vector sequence. We used the DTW for selecting key trajectory points from the selected template trajectories for feature extraction. The DTW has to align the test and prototype trajectories during each classification. Therefore, an increase in the number of gestures in the knowledge base is likely to affect the recognition performance. This also increases the computational load of classification. This problem is partially overcome by the two–stage recognition method employed in the thesis. In the first stage, unlikely gesture classes are eliminated through the maximum boundary deviation (MBD) based trajectory shape matching. The second stage of recognition involves feature vector matching through the DTW. This also enhances the overall speed of recognition.

- **Question 3:** How scalable is the approach for recognition of continuous gestures?

  Answer: Gesture spotting and movement epenthesis detection are the main challenges in continuous dynamic gesture recognition. It is difficult to distinguish between a valid gesture phase and the movement epenthesis phase only from the motion information. That is why, not many vision-based approaches for detecting motion epenthesis and co-articulation have been reported in the literature till date.

  In the proposed methods for continuous gesture recognition, we considered the cases when a particular continuous gesture sequence is composed of component gesture sequences that have similar spatio-temporal behaviour and are free of intermediate movement epenthesis. The proposed system for movement epenthesis detection and subsequent recognition of individual gestures in a continuous stream of gestures is an attempt that promises to perform well on different types of gesture sequences having different spatio-temporal characteristics and motion behaviour.

  The proposed recognizer can recognize new sentences constructed from

TH-1870_02610203

the given gesture vocabulary only if the component gestures are separat by the proposed movement epenthesis detection methods. The accuracy the whole process entirely depends on the efficient isolation of component gestures from the continuous stream of gestures. The rest of recognition is same as isolated gesture recognition. Therefore, even for long sentences, the proposed method can be applied thus ensuring scalability. We did not consider the scalability in the sense of spatio-temporal resolution and quantization levels, though the key VOP based recognition may be temporally scalable.

- **Question 4:** You have used at least one example of two-handed gesture. Can your method be easily extended to two handed gestures?

  Answer: Most of the current research on hand gesture recognition for Human-Computer Interaction deals with one-handed gestures [4], [5]. We used only two examples of two-handed gesture in connection with static hand gesture recognition. In this, we considered only the overall shape information for recognition. The other cases of static gestures where the hands can be segmented into two separate shapes, two parallel FSMs can be considered for representing two gesturing hands. In the case of static hand gestures, these parallel FSMs can be considered independently. The proposed recognition methods can only handle single-handed dynamic gestures. For two handed gesture recognition, it is important to find the self and mutual dynamics of two gesturing hands. It would be an interesting research problem to represent two handed gestures in FSM-based and trajectory-based representation. In our proposed FSM based gesture representation, it is quite possible to derive two separate FSMs for two gesturing hands. The transition of a state to the next for each hand will also depend on the state of the other hand. The incorporation of the mutual interaction between the hands is not simple and requires further investigation. Similar studies have been made for the HMM-based recognition [6]. Again, for trajectory guided recognition, we can extract two separate trajectories and associated trajectory based features for two gesturing hands. How to include mutual interaction between the gesturing hands requires further investigation.

- **Question 5:** Do you think a reject option in the continuous gesture classifier will make the system more reliable?

  Answer: The reject option is a technique used in pattern recognition for improving classification reliability. The misclassification risk can be reduced by allowing not to classify a pattern into any class. This is termed as distance reject.

  The reject option is used in the proposed gesture recognition system to reduce miss classification risk. By inclusion of reject options, the algorithm attains more adaptability to real gesture classification problems for which

TH-1870_02610203

gesture classes generally are not clearly separable and/or not completely known. Though we have not shown the rejection rates in the tables shown in Chapter 5, we used the reject option in our algorithm. The conditions of reject option is shown in the algorithm in Block Diagram 4.3. Gesture recognition results as given in Chapters 2-4 show the rejection rates corresponding to gesture classification. The recognition and rejection rates corresponding to some of the continuous gesture sequences are shown in Table 1.

Table 1
Gesture recognition rates for Type I gesture sequence.

No. of training samples per class : 18 No. of test samples per class : 32

Continuous gestures	Recognition rates	Rejection rates
One – Two – IV hand – Cartridge – Friend	96.87	0.0
One – Two – IV hand – Friend – Cartridge	96.87	0.0
One – Two – Friend – Cartridge – IV hand	84.37	6.2
One – Two – Friend – IV hand – Cartridge	87.50	3.1
One – Two – Cartridge – Friend – IV hand	96.87	3.1
One – Two – Cartridge – IV hand – Friend	96.87	0.0
One – IV hand – Two – Cartridge – Friend	90.62	3.1
One – IV hand – Two – Friend – Cartridge	90.62	3.1
One – IV hand – Friend – Two – Cartridge	87.50	0.0
One – IV hand – Friend – Cartridge – Two	96.87	0.0
One – IV hand – Cartridge – Friend – Two	84.37	6.2
One – IV hand – Cartridge – Two – Friend	81.25	9.4
One – Cartridge – IV hand – Two – Friend	84.37	3.1
One – Cartridge – IV hand – Friend – Two	84.37	6.2
One – Cartridge – Friend – Two – IV hand	84.37	3.1
One – Cartridge – Friend – IV hand – Two	84.37	0.0
One – Cartridge – Two – IV hand – Friend	96.87	0.0
One – Cartridge – Two – Friend – IV hand	81.25	6.2
One – Friend – Cartridge – Two – IV hand	96.87	0.0
One – Friend – Cartridge – IV hand – Two	96.87	0.0
One – Friend – IV hand – Two – Cartridge	90.62	0.0
Average	**90.02**	**2.5**

TH-1870_02610203

COMMENTS ON THE POINTS RAISED BY THE REVIEW

Movement epenthesis

The term "co-articulation" is replaced by "movement epenthesis" all throughout the thesis.

Advantages of the proposed system

The goal of this research work is to build a system that uses natural gestures as a modality for recognition in a vision based setup. Our main focus is to recognize dynamic hand gestures having different spatio-temporal characteristics for HCI applications. Algorithms are available for recognition of a particular type of hand gestures. There is a scope to develop a scheme that will integrate different gesture recognition methods in a common system. The research in vision-based gesture tracking and recognition is still far from maturity [5], [7]. The main challenges in this context are robustness, user independence and view independence. Vision-based recognition must achieve precision in two complementary domains – very fast tracking of the position of the hand in the 3D space and exact estimation of the configuration of the fingers in the hand. To combine these requirements in one system is still a major challenge.

The thesis aims at developing algorithms for tracking and segmentation of the hand with reduced computational load and better performance. We proposed to borrow concepts of key VOPs used in the MPEG-4 video standard for this purpose. Particularly, if the gesture sequence is MPEG-4 coded, the VOPs will be directly available for gesture modelling. As discussed in Chapter 2, the Hausdorff tracker used in our method takes care of rotation and scaling of the hand thereby saving some amount of computations necessary for determining affine motions.

The key VOP based summarization is used for representation of gestures with local hand motion only by an FSM-based method. Compared to the HMM based systems, where the number of states and the structure of the HMM must be predefined, a gesture model is available immediately in the proposed FSM based system. The major contribution here is that the proposed recognizer compares only the selected key VOPs of the input video sequence with the existing key VOPs in the prototype FSMs. This greatly enhances the recognition speed. All the finite state machine representation algorithms [8], [9], [10] use each frame in the gesture sequence to build up the gesture model. This seems to be computationally inefficient. Since there exists large amount of temporal redundancy between frames in a video sequence, gesture models obtained from only a selected number of frames in the sequence are shown to be adequate.

We compared the proposed recognition system to an HMM based recog-

TH-1870_02610203

nition system. HMM classifiers are very efficient for classifying dynar data patterns. Nonetheless, the proposed system yields recognition ac racy comparable to that by the HMM based approach.

For representation of gestures having global hand motion only, a trajectory-based method is used. Recognition rate obtained in our experiments is comparable to most other recognition methods available in the literature. As discussed in the thesis, Yang *et al.* proposed a method for extraction of 2D motion trajectories based on skin colour and subsequent gesture recognition by time delay neural network (TDNN) [11]. Since, not many vision based approaches for trajectory guided recognition have been reported for the class of gestures considered in this work, the proposed method promises to be a significant development in the field of recognition of dynamic hand gestures having only global motion.

We have developed a hand gesture recognition system that can handle different types of isolated dynamic gestures in a common platform. Most of the sign language recognizers available for different sign languages mainly deal with global hand motion in space while utilizing very limited information on local motion of the hands [11], [12]. Because of this, these systems cannot differentiate two signs having same gesture trajectory even if the hand shapes are significantly different. The gestures normally used in HCI applications contain restricted motions of the hand. This is done with an aim to increase the recognition accuracy by the available recognition techniques [4]. But, user friendly HCI applications demand for hand gestures which have both local and global motions. Most of the literature on isolated hand gesture recognition in a vision based setup has not considered the case of gestures having both these components of motion. The proposed integration utilizes the hand shape, the gesture trajectory and the hand motion information for recognition. Fusion of these different parameters in a single system enhances the versatility of the recognizer.

Another contribution of the thesis is the development of methods to tackle movement epenthesis and co-articulation in continuous hand gesture recognition. Movement epenthesis detection is one of the main challenges in continuous gesture recognition. It is difficult to distinguish between a valid gesture phase and the movement epenthesis phase only from the motion information only. The techniques developed so far for movement epenthesis and co-articulation detection are generally not useful for wide range of gesture vocabulary [13], [14], [15]. This is because motion interpretation itself is an ill-posed problem in the sense that a unique solution of the problem cannot be guaranteed [7]. Moreover, the existing algorithms do not address the problems associated with the recognition of different types of continuous hand gestures having different spatio-temporal behaviour or motion characteristics, *viz.*, gestures having only local motions, gestures having only global motions, gestures having both local and global motions, and gestures showing sequence of static hand poses (fluent finger spelling). In view of these, we propose to develop a scheme for gesture recognition that first

TH-1870_02610203

identifies the nature of hand motion involved in a continuous gesture s quence and then applies a suitable spotting method incorporating cogniti and psychological aspects of recognition in a fuzzy rule-based framework. The proposed system for movement epenthesis and co-articulation detection and subsequent recognition of individual gestures in a continuous stream of gestures is an attempt that promises to perform well on different types of gesture sequences having different spatio-temporal characteristics and motion behaviour.

Signer independence

The training and testing gesture data used for the experiment were naturally generated. The recognizer learns the wide spatio-temporal variation occurring in each gesture under consideration during the training session. The training set defines the subspace in which recognition will be performed. In our method, for each selected gesture, gesture video sequences were stored in a sequence in order of the signers. For each of these gestures, the data are sequentially partitioned into training and testing sets. Therefore, only a subset of the signers is included in the training sessions and hence most of the test data is from signers not included in the training. Thus, the test results suggest signer independence.

For example, in Table 2.4 of the thesis, 500 samples were used for training. These samples belong to three signers. The test data include 1500 samples which come from six signers not included in training and one signer from the training set.

Recognition of new sentences

The proposed recognizer can recognize new sentences constructed from the given gesture vocabulary only if the component gestures are separable by the proposed movement epenthesis detection methods. The accuracy of the whole process entirely depends on the efficient isolation of component gestures from the continuous stream of gestures. The rest of recognition is same as isolated gesture recognition. Therefore, even for long sentences, the proposed method can be applied. As an example, results on a continuous gesture sentence composed of six gestures and another composed of gestures not present in the training are shown in Table 2.

Exaggerated pauses during gesturing

In continuous gesture recognition, slight pauses are included to judge the starting and ending points of the component gestures by the proposed gesture spotting methods. But the pauses are not sufficiently long to hamper the fluency of signing.

TH-1870_02610203

Table 2
Experimental Results: Gesture recognition rates for Type I gesture sequence.

No. of training samples per class : 18 No. of test samples per class : 32

Continuous gestures	Recognition rates
Five – Two – One – Two – Friend – Two	87.50
Two – Five – Three – Seven – One	90.62

References

[1] D.P. Huttenlocher, J.J. Noh and W.J. Rucklidge, "Tracking non-rigid objects in complex scene", *Proc. Fourth International Conf. Computer Vision*, pp. 93–101, 1993.

[2] T. Meier and K. N. Nagan, "Automatic segmentation of moving objects for video object plane generation", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 525-538, 1998.

[3] B. Erol and F. Kossentini, "Automatic key video object plane selection using the shape information in the MPEG-4 compressed domain", *IEEE Trans. Multimedia*, vol. 2, no. 2, pp. 129-138, 2000.

[4] V.I. Pavlovic, R. Sharma and T.S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677-695, 1997.

[5] S.C.W. Ong and S. Ranganath, "Automatic sign language analysis: A survey and the future beyond lexical meaning", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 873-891, 2005.

[6] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition", *Proc. Computer Vision and Pattern Recognition (CVPR)*, 1997.

[7] Y. Wu and T. S. Huang, "Hand modelling, analysis, and recognition for vision based human computer interaction", *IEEE Signal Processing Magazine*, pp. 51-60, 2001.

[8] A.F. Bobick and A.D. Wilson, "A state-based approach to the representation and recognition of gesture", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, pp. 1325-1337, 1997.

TH-1870_02610203

[9] J. Davis and M. Shah, "Visual gesture recognition", *Vision, Image and Sig Processing*, vol. 141, no. 2, pp. 101-106, 1994.

[10] P. Hong, M. Turk and T.S. Huang, "Gesture modelling and recognition using finite state machines", *Proc. IEEE Conf. Face and Gesture Recognition*, pp. 410-415, 2000.

[11] M. H. Yang, N. Ahuja and M. Tabb, "Extraction of 2D motion trajectories and its application to hand gesture recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1061-1074, 2002.

[12] E. J. Holden, G. Lee and R. Owens, "Australian sign language recognition", *Machine Vision and Applications*, vol. 16, no. 5, pp. 312-320, 2005.

[13] C. Vogler and D. Mextaxas, "Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods", *Proc. IEEE International Conf. Systems, Man and Cybernetics*, pp. 156-161, 1997.

[14] P. Morguet and M. Lang, "Spotting dynamic hand gestures in video image sequences using hidden markov models", *Proc. International Conf. Image Processing*, pp. 193-197, 1998.

[15] A. Shamaie and A. Sutherland, "Classification of co-articulated and partly occluded hand gestures", *Proc. Irish Machine Vision and Image Processing Conf.*, 2001.

621.3
BHU/V
P06

THESIS
Lakshminath Bezbaroa Central Library
Indian Institute of Technology Guwahati
Acc. No. TH...1870...
Date.....28/3/12.....

TH-1870_02610203