
Static and Dynamic Interpretations of Hand Specific Body Language Cues for HCI

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

von M.Sc. Omer Rashid Ahmad

geb. am 20.03.1983 in Riad, Saudi-Arabien

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik

der Otto-von-Guericke-Universität Magdeburg

Gutachter:

Prof. Dr.-Ing. habil. Ayoub Al-Hamadi

Prof. Dr. rer. nat. Andreas Wendemuth

Prof. Dr. rer. nat. Heiko Neumann



FAKULTÄT FÜR
ELEKTROTECHNIK UND
INFORMATIONSTECHNIK

Promotionskolloquium am 07.12.2015

Acknowledgments

I would like to thank my supervisor Prof. Dr.-Ing. habil. Ayoub Al-Hamadi for his guidance and relentless support over the course of this PhD. His commitment of research inspires me over all these years and motivates me to do my best. I am grateful to Prof. Dr. rer. nat. Andreas Wendemuth and Prof. Dr. rer. nat. Heiko Neumann for accepting to review my thesis.

I would like to thank my colleagues in NIT research group, particularly, Dr.-Ing. Gerald Krell and Sebastian Handrich for their countless stimulating conversation and invaluable technical advices. Special thanks to my family for their infinite support and understanding.

Abstract

Interaction with machines using vision-based technologies is a vital research domain in which the body language cues plays an attributed role in the communication. In this thesis, body language cues particularly hand gesture and posture have been investigated to recognize, interpret and infer meaningful expressions while interacting with computers. The main aim of the thesis is to propose an intuitive interactive interface with real-time operability and robust performance without encoding the spatial hardware or fiducials to ensure higher level of ease, flexibility and naturalness.

The research in this thesis is fragmented into two main parts. In the first part, a framework consisting of repertoire of algorithms has been proposed starting with the segmentation where normal Gaussian distribution is used to cluster the skin blobs (i.e., face and hand blobs) from the image streams by utilizing the depth observations. Further, the face blobs are detected using Haar-like features and are used for online training purpose when the segmentation fails. Moreover, hand blobs are processed using distance transformation descriptor to eliminate arm-region thus giving the refined hand blobs. Afterwards, features are extracted to recognize hand gestures and postures in which for hand gesture recognition, Bezier descriptors are constructed by transforming the hand centroid points into a set of Bezier points. Later, Bezier descriptors are formed by determining the difference between consecutive Bezier points which are then quantized and concatenated. In the hand posture recognition, fingertips are detected from the detected hand blob using the curvature features and is used as a criterion to categorize the hand posture symbols into groups to reduce the mis-classifications among posture symbols. Further, statistical and geometrical features are extracted, analyzed and integrated for hand posture recognition. To maintain the identities of the hand blobs, Iterative Closest Point algorithm is employed for objects tracking using Fast Library for Approximate Nearest Neighbors tracker spatially over time. The extracted features are finally given to the classifier for hand gesture and posture symbol recognition using Hidden Markov Model and Support Vector Machines respectively. The classification outcomes of hand gesture and posture recognition provide a fundamental basis for the integration of hand gesture and posture modalities. A new approach is proposed by incorporating a Particle-filter system that computes the contribution weights of these

modalities for the integration and by doing so, the ambiguities observed in the classification process is addressed. The integration of hand gesture and posture recognition leads to deduce interpretations and derives the inferences using Context Free Grammar rules.

In the second part of thesis, new methodologies are proposed to augment the virtual components over hand which utilizes the developed algorithms for hand gesture and posture recognition. This work begins by extracting the features from segmented hand and spatial relationship is built among the extracted hand components (i.e., hand palm and fingertips). To do this, path derivation process is proposed to acquire the optimal path from fingertips to palm center by incorporating the distance scores and segmented skin pixels. In this way, a structural representation is constructed in the form of skeleton which mimics the actual hand physics. Afterwards, the patches are formed from the hand skeleton points by building the correspondence of two detected neighboring fingers on which the pose is estimated. The individual extracted pose parameters are finally aggregated to augment the *3D* objects on the hand.

The algorithms proposed in this thesis are extensively tested where the hand gesture and posture classification result in 98.3% and 97.8% recognition rate respectively. Moreover, integration of these modalities results in 98.6% recognition rate for the meaningful expressions which proves the significance of the proposed approach. In the second part, the experiments are conducted to evaluate the performance of pose estimation and augmentation. Moreover, the comparative analysis is carried out on different patch models and marker-based fiducial detection approach where the re-projection error is measured for the estimated pose.

Zusammenfassung

Die Mensch-Maschine Interaktion unter Verwendung bildgestützter Technologien stellt einen wichtigen Forschungsschwerpunkt dar, in welchem die Erkennung der Körpersprache eine unverzichtbare Rolle einnimmt. In dieser Arbeit werden Elemente der Körpersprache, insbesondere Handbewegungen und Handstellungen hinsichtlich der Erkennung, Interpretation und Ableitung bedeutungsvoller Ausdrücke während der Mensch-Maschine Interaktion untersucht. Das Ziel besteht in der Schaffung eines intuitiven, bildgestützten HCI Systems, welches eine echtzeitfähige und robuste Erkennungsleistung ermöglicht und hierbei, um die Einfachheit, Flexibilität und Natürlichkeit der Interaktion zu gewährleisten, auf die Verwendung von Markern und spezieller Hardware verzichtet.

Der Forschungsgegenstand dieser Arbeit ist in zwei Teile untergliedert. Im ersten Teil wird ein Framework bestehend aus einem Repertoire von Algorithmen zur Erfassung und Klassifikation der Handbewegung und Handstellung vorgeschlagen. Der erste Schritt besteht in der Segmentierung hautfarbener Bereiche (d.h. des Gesichts und der Hände). Hierzu werden hautfarbene Bereiche mittels eines auf der Gauss-Verteilungen basierenden Modells in Farb- und Tiefenbildsequenzen geclustert. Die Trainingsparameter des Modells lassen sich im Falle eines Fehlschlagens der Segmentierung anhand der mittels Haarlike Features detektierten Gesichtsregion zur Laufzeit neu adaptieren. Durch eine Beschreibung der segmentierten Handbereiche mit auf der Distanztransformation basierenden Deskriptoren, wird die Position der Handfläche bestimmt und der unerwünscht segmentierte Bereich des Unterarms entfernt. Die hieraus erzielte verbesserte Handsegmentierung ermöglicht die Extraktion robuster Merkmale für die Klassifikation der Gestik und Handstellung.

Im Bereich der Gestenerkennung stellen wir einen neuwertigen Ansatz basierend auf Bézier-Deskriptoren vor. Diese werden aus der Verkettung quantisierter Bézierpunkte gebildet, welche anhand einer Modellierung der Handmittelpunkte durch Bézierkurven bestimmt werden. Für die Klassifikation der Handhaltung wird ein Ansatz vorgeschlagen, welcher zunächst auf Basis von Krümmungsparametern der Handkontur die Fingerspitzen detektiert. Anzahl und Ort erkannter Finger legen Kriterien für eine Unterteilung der zu erkennenden Handstellungen in Untergruppen fest, wodurch sich eine Reduktion der Fehlklassifikation der verschiedenen Handstellungen erzielen

lässt. Die Merkmalsvektoren für die Klassifikation der Handstellung werden aus der Extraktion und Integration geometrischer und statistischer Merkmale gebildet. Es wird zudem der Iterative-Closest-Point Algorithmus in Kombination mit Fast Library for Approximate Nearest Neighbors verwendet, um während auftretender Hand/Gesichts- bzw. Hand/Hand- Verdeckungen eine zeitliche Verfolgung der Hände beizubehalten. Die extrahierten Merkmale werden abschliessend mittels entsprechender Klassifikatoren für die Erkennung der Gestik und Handstellungen klassifiziert. Hierzu wurden Hidden-Markov-Modelle bzw. Support-Vector-Maschinen verwendet. Das Ergebnis der Klassifikation der Gestik und Handstellung ist eine Menge klassifizierter Symbole dieser beiden Modalitäten. Ein nächster logischer Forschungsschwerpunkt dieser Arbeit besteht somit in der Entwicklung eines Verfahrens zur Fusionierung beider Modalitäten, um hieraus bedeutungsvolle Ausdrücke der Interaktion ableiten zu können. Zu diesem Zweck wird ein neuartiges Partikelfilter-basiertes Fusionssystem vorgeschlagen, welches Gewichtungen festlegt, mit denen beide Modalitäten kombiniert werden. Basierend auf der Integration und Interpretation der kombinierten Modalitäten werden bedeutungsvolle Ausdrücke der Interaktion bestimmt und entsprechend den Regeln einer kontextfreien Grammatik interpretiert.

Im zweiten Teil dieser Dissertation werden neue Konzepte zur Augmentierung der virtuellen Komponenten abgeleitet und entwickelt, indem einige der zuvor zur Handgesten und -Haltungen entwickelten Konzepte zur Posebestimmung und Augmentierung eingesetzt werden. Anhand dieser Merkmale wird unter Verwendung einer vorgeschlagenen Methode zur Ermittlung der optimalen Pfade zwischen Fingerspitzen und Handfläche auf die räumliche Anordnung der einzelnen Finger in Bezug zur Handfläche geschlossen. Ausgehend von dieser somit erhaltenen strukturellen Beschreibung der Hand in Form eines Skeletts werden die Lage und Orientierung von zwischen zwei jeweils benachbarten Fingern aufgespannten Flächen bestimmt. Durch eine Integration über alle Flächen wird die finale Pose der Hand ermittelt und zur virtuellen Überlagerung eines 3D-Objekts mit der Hand verwendet. Die vorgestellten Verfahren wurden ausführlich getestet und deren Leistung hinsichtlich Klassifikationsrate, Rechenzeit, Robustheit in unkontrollierten Umgebungen und Nutzerfreundlichkeit bewertet.

Im ersten Teil werden Experimente bzgl. der Erkennung der Gesten und Handhaltungen unter Verwendung verschiedener Merkmale durchgeführt und

deren Performanz untersucht. Für die Gestenerkennung wurde eine Klassifikationsrate von 98,3% und für die Handhaltung eine Klassifikationsrate von 97,8% erreicht. Es wurden weiterhin Experimente für den genannten Ansatz zur Bestimmung bedeutungsvoller Ausdrücke durch eine Integration beider Modalitäten (Gestik und Handhaltung) durchgeführt und hierbei eine Erkennungsrate von 98,6% erzielt, aus der sich die Signifikanz des vorgeschlagenen Ansatzes entnehmen lässt. Die im zweiten Teil vorgestellte Methode zur flächenbasierten Bestimmung der Handpose für die Verwendung in einer Augmented Reality Umgebung wurde ebenfalls in Experimenten evaluiert. In einer vergleichenden Analyse wurden hierbei verschiedene Flächenmodelle mit einem marker-basierten Verfahren quantitativ verglichen, wobei als Fehlermass die Abweichung der Rückprojektion der Handpose von den Markerpositionen verwendet wurde.

Dedication

*To my parents for envisioning their dreams in me,
To my family for supporting in the persuasion of their dreams, and
To my wife Saira and daughter Maryam for being my motivation and life.*

Declaration

I, hereby declare that the presented work is done without undue assistance from third parties and have made no use other than the indicated resources directly or indirectly.

Some parts of the work presented in this thesis have been published in international conferences and journals (in publication list).

Magdeburg, 27. April 2015

Omer Rashid Ahmad

Contents

List of Figures	xix
List of Tables	xxiii
Glossary	xxiv
Nomenclature	xxvii
1 Introduction	1
1.1 Motivation	2
1.2 Concept Definition	3
1.3 Contributions	6
1.4 Organization of the Thesis	7
2 Literature Review	9
2.1 Fundamental Concepts	9
2.2 Vision-based Hand Cues	10
2.2.1 Model based Approaches	11
2.2.2 Appearance based Approaches	11
2.2.3 Discussion	13
2.3 Gesture Recognition	13
2.3.1 Discussion	16
2.4 Posture Recognition	18
2.4.1 Discussion	21
2.5 Augmented Reality	23
2.5.1 Augmentation	24
2.5.2 Discussion	25
2.6 Summary and Conclusion	26
3 Segmentation	27
3.1 Context Description	27
3.1.1 Context for Gesture and Posture Scenario	28
3.1.2 Context for Augmented Reality Scenario	29
3.2 Skin Color Segmentation	29

3.3	Blob Detection	31
3.3.1	Face Blob	32
3.3.2	Hand Blobs	34
3.4	Refinement using Distance Transformation Descriptor	35
3.5	Experimental Results and Analysis	37
3.5.1	Gesture and Posture Recognition Scenario	38
3.5.2	Augmented Reality Scenario	40
3.5.3	Analysis	40
3.6	Summary and Conclusion	41
4	Feature Extraction and Tracking	43
4.1	Features	43
4.2	Gesture Features	44
4.2.1	Bezier Descriptors	44
4.2.2	Features Binning	48
4.3	Fingertip Detection	49
4.4	Posture Features	53
4.4.1	Statistical Feature Vectors	55
4.4.2	Geometrical Feature Vectors	57
4.4.3	Categorization based on Fingertip Detection	58
4.4.4	Experimental Results	59
4.5	Feature-Level Fusion for Posture Features	61
4.6	Tracking	62
4.7	Summary and Conclusion	67
5	Classification	68
5.1	Hidden Markov Models	68
5.1.1	Elements of HMM	68
5.1.2	HMM Topologies	70
5.1.3	Problems of HMM	72
5.1.4	Solution to Problems	72
5.1.5	Experimental Results	76
5.2	Support Vector Machines	82
5.2.1	Categorization Results and Analysis	84
5.2.2	Experimental Results	86
5.3	Summary and Conclusion	89

6	Integration and Inferences	95
6.1	Concept	95
6.2	Particle Filter System	96
6.2.1	Initialization	97
6.2.2	Selection	98
6.2.3	Prediction	98
6.2.4	Update	98
6.3	Interpretation and Inferences	100
6.4	Experimental Results and Analysis	104
6.5	Summary and Conclusion	106
7	Content Augmentation over Hand Postures	109
7.1	Hand Skeleton Formation	109
7.2	Hand Posture Geometry	111
7.2.1	Patch Detection	111
7.2.2	Content Augmentation	113
7.3	Experimental Results and Analysis	114
7.4	Summary and Conclusion	117
8	Summary and Future Directions	118
8.1	Summary	118
8.2	Future Directions	120
A	Appendix	122
A.1	Orthogonal Moments	122
A.1.1	Zernike Moments:	123
A.2	Experimental Setup	124
	Bibliography	125

List of Figures

1.1	Categorization of the body language cues as eye movements, facial expressions, body postures and gestures.	2
1.2	Conceptual model of spatial and semantic inferences.	3
2.1	Types of vision-based hand cues: model based and appearance based approaches	10
2.2	Conceptual representation of a gestural action (i.e., digit 2) in a continuous stream.	12
3.1	Proposed skin segmentation framework.	27
3.2	Ideal case of segmentation	30
3.3	Original images and skin color segmentation on long and short sleeve images.	31
3.4	a) Face images with Y, C_b and C_r channels. b) Histogram of C_b and C_r channel. c) Gaussian fitted on C_b and C_r channel.	32
3.5	Ideal and non-ideal case of skin color segmentation	33
3.6	Distance transformation of the images	36
3.7	Gesture and pose scenario for blob detection	38
3.8	Augmented Reality scenario for blob detection	39
4.1	Gesture and posture recognition: Extraction of gesture and posture features with analysis.	44
4.2	Gesture stream with detected hand centroid and Bezier points	45
4.3	Gesture stream with hand centroid and Bezier points	46
4.4	The drawing patterns of hand gestural symbols for alphabets and numbers.	47
4.5	Training samples of <i>Gesture '7'</i> and <i>Gesture '9'</i> used in classification.	48
4.6	Gestural action formation for <i>Gesture '9'</i>	49
4.7	Gestural action formation for <i>Gesture '7'</i>	50
4.8	Image with detected fingertips in IIKT-GP dataset.	52
4.9	Image with detected fingertips in IIKT-AR dataset.	53
4.10	The standard ASL finger-spelling alphabets and numbers.	54
4.11	Image sequence presents hand posture sign 'A' along with features	59

4.12	Image sequence presents hand posture sign ‘B’ along with features	60
4.13	Sequence of gestural action where the subject is drawing <i>Gesture</i> ‘K’ with observed occlusion	65
4.14	Sequence of gestural action where the subject is drawing <i>Gesture</i> ‘K’ with graphs.	66
5.1	Ergodic model	69
5.2	Left-Right Model	70
5.3	Left-Right Banded Model	71
5.4	Gestural action when the subject is drawing <i>Gesture</i> ‘8’ using Bezier descriptor ($N = 15$)	77
5.5	Gestural action when the subject is drawing <i>Gesture</i> ‘8’ using Bezier descriptor ($N = 5$)	78
5.6	Gestural action when the subject is drawing <i>Gesture</i> ‘8’ using Bezier descriptor ($N = 10$)	78
5.7	Gestural action when the subject is drawing <i>Gesture</i> ‘8’ using Bezier descriptor ($N = 20$)	79
5.8	Gestural action when the subject is drawing <i>Gesture</i> ‘8’ using Bezier descriptor ($N = 30$)	80
5.9	Classification of Bezier descriptors (i.e., $N = \{5, 10, 15, 20, 30\}$) using HMM with recognized <i>Gesture</i> ‘8’ and ground truth. . .	80
5.10	Confusion matrices of gesture symbols.	81
5.11	Classification rate of Bezier descriptors and centroid points using HMM.	82
5.12	a) Margin of the hyper-plane. b) Mapping from input data to a richer feature space through kernel function.	83
5.13	Sample images from sequence presented for hand posture signs ‘C’ and ‘L’ along with features and classification.	87
5.14	Classification rate of posture symbols with and without fingertip detection process.	90
6.1	Process flow of the proposed framework for integration, interpretation and inference.	96
6.2	Particle filter process: initialization and updation	99
6.3	Meaningful expression “Seven Banana Juices” results from recognized ASL posture symbols ‘B’ followed by ‘A’ to result in ‘Banana’ and gesture ‘7’.	107

6.4	Meaningful expression “Three Apple Juices” results from recognized gesture symbols ‘A’ and ‘P’ and the posture symbol ‘3’	108
7.1	The framework for augmenting the virtual contents on hand postures.	110
7.2	Original image with distance transformation and patch description.	110
7.3	Distance transformation and path derivation for hand skeleton.	111
7.4	Hand palm and fingertip detection along with the fingertip clustering.	113
7.5	Results of content augmentation on fiducial marker.	114
7.6	Results of content augmentation on hand postures.	115
7.7	Bezier features extraction and classification results on image sequences with 1 fingertip detected.	116
A.1	a) First Context (IIKT-GP): Kinect camera is oriented in front of the subject for hand gesture and posture recognition. b) Second Context (IIKT-AR), webcam is adjusted in front of the subject in a tilted manner (i.e., 45 orientation) for AR scenario.	124

List of Tables

3.1	Precision and Recall: Hand Detection and Palm Center with or without Distance Transformation (DT)	41
4.1	Confusion Matrix: Fingertip Detection	54
4.2	Fingertip Detection	61
4.3	Feature Combinations for Posture Analysis	62
5.1	Average processing-time in milliseconds (640×480) for the proposed approach of Gesture (G) and Posture (P) recognition . .	89
5.2	Confusion Matrix of Statistical and Geometrical Features - Group 1 with no Fingertip Detected	91
5.3	Confusion Matrix of Statistical and Geometrical Features - Group 2 with One Fingertip Detected	92
5.4	Confusion Matrix of Statistical and Geometrical Features - Group 3 with Two Fingertips Detected	93
5.5	Confusion Matrix of Statistical and Geometrical Features - Group 4 with Three Fingertips Detected	94
6.1	Lexicon of Symbols	100
7.1	Average processing-time in milliseconds (640×480) for different modules of the proposed approach	112
7.2	Comparison between Fiducial Detection and Point Models (Mean Re-projection Error)	114

Glossary

AR Augmented Reality.

ASL American Sign Language.

BSL British Sign Language.

BW Baum-Welch.

CFG Context Free Grammar.

CSL Chinese Sign Language.

DOF Degrees of Freedom.

DT Distance Transformation.

DTW Dynamic Time Warping.

EM Ergodic Model.

FLANN Fast Library for Approximate Nearest Neighbors.

FSL French Sign Language.

Fuzzy K-NN Fuzzy K-Nearest Neighbor.

GAL Gaining Algorithm Learning.

GSL Greek Sign Language.

HCI Human Computer Interaction.

HMM Hidden Markov Models.

HOG Histogram of Gradients.

ICP Iterative Closest Point.

ISL International Sign Language.

K-NN K-Nearest Neighbor.

LRBM Left-Right Banded Model.

LRM Left-Right Model.

MvMF Mixture of Mises-Fisher.

PCA Principal Component Analysis.

PDF Probability Density Function.

PF Particle Filter.

ROI Region of Interest.

SPDS Single Parameter Dynamic Search.

SVD Singular Valued Decomposition.

SVM Support Vector Machines.

TOF Time of Flight.

UCF University of Central Florida.

Nomenclature

α_{gstr}	Contribution-weights of Gesture
α_{pstr}	Contribution-weights of Posture
κ	Curvature
\mathcal{I}	Integration
\mathcal{R}_{hmm}	Classification outcome of Gesture
\mathcal{R}_{svm}	Classification outcome of Posture
$\mathfrak{P}(\mathbf{x})$	Probability of pixel \mathbf{x}
μ	Mean
$\phi_{1...7}$	Hu-Moment features
Σ	Covariance
σ	standard deviation
\mathbf{d}_i	Distance Vector
φ	Discrete Vector
ξ	Pose
ζ	Patch
a_{ij}	Transition Matrix
AC	Active Region
B_d	Bezier Descriptor
b_{it}	Observed Symbols Matrix
BP	Bezier Points
C_s	Contour Segment

Circ Circularity

CP Control Points

$d(p, q)$ Euclidean distance between points p and q

FT Fingertip

Geo_{pstr} Geometrical features set for posture

Gstr Gesture Feature set

$K(x, y)$ Kernel Fuction

nD Normalized Distance

$O = \{o_1, o_2, \dots, o_T\}$ Set of Observations

PA Passive Region

Pstr Posture Feature set

Qf Quantized features

R_p Representative Points

Rect Rectangularity

ro Rotation

$Stat_{pstr}$ Statistical features set for posture

tr Translation

Introduction

Daily life communication is dominated by verbal information exchange but the non-verbal cues are undoubtedly equally significant, as it reflects the mental state of a person. Indeed, non-verbal cues enable the abstract interpretation of body language to express, convey and exchange the emotional conditions as a standalone communication medium even when the verbal information is not sufficient. Various psychologists advocate that the *body language cues* contribute about 50 – 70% in the whole communication process [1]. Of the four non-verbal body language cues - facial expressions [2, 3], gaze expressions like eye movements [4], body postures [5, 6] and gestures [7, 8] as shown in Fig. 1.1, the body postures and gestures are superior to other body language cues because of its vivid visibility and distinctive appearances even at a distance. In contrast, the facial (i.e., happiness, sadness, surprise etc.) and gaze expressions (i.e., thinking, blinking eyes etc.) have no obvious deductions due to limited visibility in wide field of view. Practically, hand gestures have a great variety of applications compared to the body postures (i.e., pose, gaming applications, sports analysis, skeleton models etc.) which includes the communication in social gatherings [9], traffic management signs [10], gaming environments [11, 12], and most recently used to interact with the machines and smart devices [13]. Moreover, these body language cues influence and reflect the correspondence with the context and surrounding in which they are being performed.

In this technological era, man is surrounded by variety of machines from high processing computers to smart hand-held devices and expecting these machines to act and function autonomously. Specifically, the interaction methodologies have been revolutionized significantly from its earlier form (i.e., command like interfaces such as mouse and keyboards etc.) and the focus is diverted to exploit the natural interaction medium that is, *body language cues*. Researchers from various disciplinary fields such as computer graphics, visualization and computer vision are envisioning new methodologies to

Body Language	Eye Movements	Facial Expressions	Body Postures	Gestures
Example	<ul style="list-style-type: none"> • Open/Closed • Eye Rolling • Focusing 	<ul style="list-style-type: none"> • Happy/Sad • Anger/Surprise • Disgust/Fear 	<ul style="list-style-type: none"> • Skeleton • Gaits • Pose 	<ul style="list-style-type: none"> • Hand Gestures • Hand Postures • Hand Actions
Application Areas	<ul style="list-style-type: none"> • Click/Browse • Fatigue Detection • Gaze Analysis 	<ul style="list-style-type: none"> • Sign Languages • Facial Actions • Emotions 	<ul style="list-style-type: none"> • Gait Identification • Pose Detection • Tracking/Sports 	<ul style="list-style-type: none"> • Sign Languages • Gesture/Posture • Command/Control • Tracking

Figure 1.1: Categorization of the body language cues as eye movements, facial expressions, body postures and gestures.

design immersive interfaces to interact with the machines. Consequently, the scope of *human computer interaction (HCI)* [14, 15, 16] has been broadened to visual interfaces where the interaction with machines is performed through visual sensors (i.e., interaction with smart devices, interaction with virtual and augmented contents, gaming applications, command and control interfaces etc.) given the body language cues as an input medium. Therefore, in the domain of HCI, computer vision has played an attributed role with the ambition to allow humans to interact through body language cues virtually and feel the interactivenss [17] which is not possible through traditional HCI devices (i.e., mouse and keyboards). This technological advancement is continuously spurring the virtual interfaces utilizing body language cues to transform the living environment into a digital smart world (e.g., command and control of machine and robots through gestures, control by scrolling through eyes etc.). However, the fundamental criteria in the design of interactive interfaces for the body language cues is the naturalness and intuitiveness which has been violated by the introduction of special devices such as hand gloves, markers or control units attached to the human body [18, 19]. In this thesis, a repertoire of novel approaches has been proposed utilizing hand gesture and posture cues to design a robust and efficient visual interaction interface for real-time scenarios where the key priority is to maintain the user's ease and flexibility by satisfying the criterion of naturalness and intuitiveness.

1.1 Motivation

One of the biggest challenges of computer vision is the ability to interact with the machines and users in an un-constrained environment. Particularly,

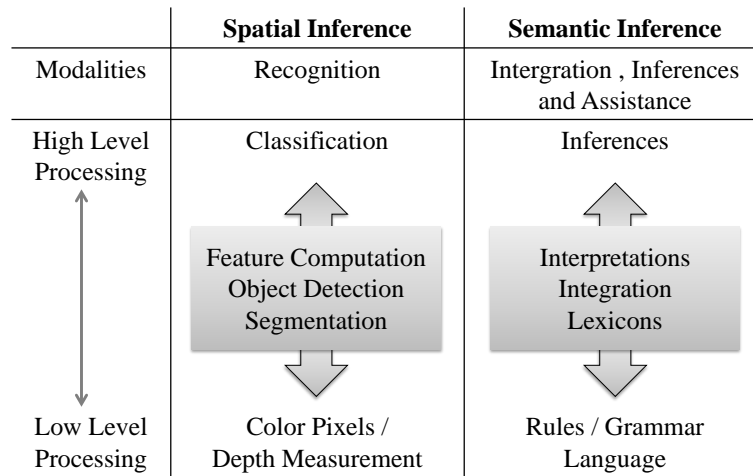


Figure 1.2: Conceptual model of spatial and semantic inferences.

vision-based HCI system should be capable of performing the real-time interaction with natural response-time; which is the motivating factor of this research and is indeed one of the challenging issues to be addressed in HCI domain.

Despite of recent advancement in vision-based HCI system, reasonable working assumptions are taken into account which provides the basic building blocks for the research and development. For instance, in many applications, researchers [20, 21, 22, 23] have used body fiducial markers, specific body cues (i.e., up and down), special clothes, and hand gloves (i.e., with and without colored markers) to acquire accurate data and to ensure higher certainty. Nevertheless, the assumptions such as ease of recognition using marker-based approaches and specialized gloves or clothes to avoid the segmentation issue; leads to vision-based HCI system that doesn't fulfill the criteria of naturalness and ease. The need is to develop algorithms which are robustly performing the interpretation, recognition of hand gesture and posture body cues without embodying the spatial hardware or fiducials ensuring the higher level of ease, flexible and naturalness.

1.2 Concept Definition

The human computer interaction gets its strength from the building blocks of human thought process. Based on this naturally inspired concept, the struc-

ture of HCI system is constructed and broadly categorized in two main types: 1) Direct Interaction and 2) Indirect Interaction; where the interaction is performed through various modalities (i.e., visual, audio, haptic etc.). In direct interaction, the subject is directly interacting with machines (i.e., interaction with machines using hand gestures or body postures etc.) whereas in indirect interaction, the subject is interacting via an object (e.g., writing on the board using pen, playing golf etc.). In the indirect interaction, normally the object's knowledge (i.e., pen and board etc.) is necessary for building the inference model about the underlying activities in the scene. But, the main goal of the system remains the same which is to recognize body cues of the subject during direct or indirect interaction in real-environment.

The interaction is performed with machines through single or multiple interacting subjects and *work flow* defines the interpretation process of HCI system which is guided by high level process to define the collaborative lexicon, protocols and their intended outcome. In the interaction scenario, the main queries to be answered are:

- How many subjects exist in the scene?
- What kind of body language cues is the subject utilizing for interaction?
- How to process the body language cues to transform the raw data into information patterns?
- How to develop the generalized classification scheme to recognize the body language cues commands or symbols?
- How to develop the lexicons and rules for the interpretation and inference?
- What kind of assistance is provided by HCI interface (i.e., Augmented Reality (AR) or Virtual Reality (VR) etc.)?

These outlined queries lead us to recognize, interpret and infer the body language cues for vision-based HCI system comprising of spatial and semantic inferences as shown in Fig. 1.2. In this thesis, the spatial inferences involve the segmentation, identification and labelling of objects (i.e., hands and face), their association and classification. However, the semantic inferences resolve the ambiguities associated with the classified patterns, the interpretation derived from the non-dubious patterns, generation of meaningful expressions

from different modalities and their integration. Moreover, both spatial and semantic inferences are the high level inference units built upon the low level algorithms like the color pixels and depth measurement, and grammar rules respectively. We are also intended to examine what kind of assistance is supported with HCI interface (i.e., embedding AR based information over hand postures). In this thesis, we are aspired to identify and recognize the underlying spatial and semantic inferences in the static and dynamic scenes during the interaction with a subject.

The conceptualization of spatial inference for hand gesture and posture body cues as shown in Fig. 1.2 takes its input from the low level algorithms as raw data in the form of image and depth streams. These streams are input to segmentation process to categorize the interesting (i.e., skin pixels) and non-interesting pixels. Further, the feature maps are built for both gesture and posture modalities. This is indeed a higher level of processing where different classes or symbols are created and analyzed for the gesture and posture recognition. These classes are finally recognized using features in the classification process to discriminate the symbols. The semantic inference takes its input from the classified outcomes of hand gesture and posture symbols. The grammar rules are developed to generate the interpretations for hand gesture and posture modalities. The generated interpretations are finally used to infer the meaningful expressions from the hand gesture and posture modalities using the set of lexicon and context free grammar rules. Similarly, the virtual contents are overlaid over the hand postures in AR application for the assistance.

In this thesis, the research is conducted considering two different applications scenarios. In the first scenario, a novel framework is proposed for understanding the gesture and postures body cues as an interaction medium for machines where the subject is standing in front of the vision sensor. In the second, a new 3D interactive interface is proposed where the virtual components (i.e., 3D objects) are overlaid based on inferred geometry from hand postures in AR application. Both of these scenarios utilize similar algorithms for low-level image processing which includes segmentation and feature detection and further on, due to varied nature of objectives, separate algorithms are proposed to accomplish the desired research goals as mentioned in the following section.

1.3 Contributions

Challenges are tough but when addressed, they become contributions. In this thesis, the achieved contributions are presented below.

Foreground Objects: The extraction of important contents (i.e., foreground or object of interest) from raw data is a pertinent step for analyzing the underlying scene and is still challenging. In this thesis, the foreground objects (e.g., face and hands) are detected and tested under different lighting conditions, complex background, and various ethnicities. However, the criterion of performance is based on the robustness of developed methods under different real-situations. In addition, a training mechanism is proposed at run-time which stabilizes the hand and face detection process when the skin contents are not detected correctly.

Occlusion: In the scene, the complex interactions result in the frequent occlusions among the detected objects (i.e., in our case, the hands and face) which makes the measured features such as, detected interest points, and the classifiers result fairly unreliable. To maintain the object's identities, Iterative Closest Point (ICP) algorithm is employed for tracking using Fast Library for Approximate Nearest Neighbors tracker to track the motion of these objects (i.e., hands and face) spatially.

Features Extraction and Categorization: Features are the backbone of any framework as it contributes directly to the recognition process. In this context, an important attribute of any feature is its invariance to translation, rotation and scaling parameters. Therefore, in this thesis, an important contribution lies in the selection of distinct features (i.e., Bezier features, moments, geometrical features etc.) for hand gestures and postures to enhance the recognition rates based on extensive analysis in unconstrained environments. Besides, the overall performance can be significantly improved by the categorization of symbols prior to classification. In this thesis, an approach is suggested to categorize the posture symbols through fingertip detection process which greatly improves the classification rate.

Classification: The classification process recognizes the underlying symbols

by taking features as input. In this thesis, we have paired different features together with and without the categorization process to see the performance of hand gesture and posture recognition. Moreover, Hidden Markov Model (HMM) and Support Vector Machines (SVM) are used for the classification of hand gesture and posture recognition respectively.

Integration, Interpretation, and Inference: Fusion of different features helps to recognize the meaningful expressions generated from different modalities and discard the dubious features. Dubiety can occur due to the low recognition rates or for the symbols having very similar shape and structure. In this thesis, a particle filter system is proposed to disambiguate the uncertainty involved in the symbol recognition process. Moreover, we have designed a lexicon for recognized symbols from gesture and posture modalities and define the grammar rules to extract the meaningful expressions.

Virtual Content Augmentation over Hand Posture: Virtual content augmentation over hand postures without using special markers is a challenging task. As, we argue over the natural and intuitive interface, therefore, in this thesis, we propose a method to determine the pose over hand postures and overlays the virtual contents on it. To achieve this, hand skeleton model is built based on the detected features (i.e., using hand palm and fingertip detection etc.).

1.4 Organization of the Thesis

This chapter presented the gesture and posture recognition problem, described the key motivations and overall goals of this thesis. The outline is structured as:

Chapter 2: In this chapter, state of the art approaches are categorized according to the adapted research strategy for gesture, posture and hand-based augmented reality. In this chapter, we aspire to categorize the reviewed literature based on the methodologies used to develop the solutions, provided a detail description of representative methods in each category, and examined their pros and cons.

Chapter 3: This chapter presents the hand and face segmentation problem from the background as a pertinent requirement to perform object detection

and feature extraction for gesture and posture modalities. It also reflects the underlying diversified issues which appears with the segmentation of hands and face.

Chapter 4: In this chapter, the feature extraction for the gesture and posture modalities are presented by incorporating the global and local features respectively. Moreover, the occlusion is handled through an (ICP) algorithm which takes the local features as the observation and resolve the ambiguities between the hands and face to maintain the tracking process.

Chapter 5: In this chapter, the classification scheme is presented for the gesture and posture modalities to evaluate the performance of the proposed approach. The experimental results are presented on IIKT-GP dataset for the gesture and posture modalities along with the comparative analysis and evaluation.

Chapter 6: This chapter presents the concept of different levels of feature fusion for the integration of gesture and posture modalities, focussed on the interpretation and inferences rules.

Chapter 7: In this chapter, the virtual components are augmented over the computed hand postures. The hand skeleton is built by detecting and linking the hand physical components (i.e., using palm and fingertip detection) for determining the pose. The experimental results are presented on IIKT-AR dataset for the hand postures and augmentation along with comparative analysis and evaluation.

Chapter 8: Chapter 8 concludes this thesis with a summary and the description of future directions.

Literature Review

In recent years, body language cues get huge attention in vision based HCI domain due to its natural way of interaction with machines. This motivates the researchers to investigate new methodologies incorporating body language cues as interaction mediums for variety of applications. Among these body language cues, hand gestures and postures cues have advantages due to their distinctive visibility and wide usability. This entails a vision based HCI system to be designed with high flexibility, natural, and robust interaction interface. This chapter is dedicated to describe the fundamental concepts in the domain of gesture and posture recognition, reviews the state of art, outlines the research gaps and finally pinpoints the objectives of the research by providing the pros and cons of the defined approaches. This chapter is sectioned as: Section 2.1 provides the fundamental concepts to build the basic understanding of the proposed research domain, Section 2.2 presents the vision based hand cues, Section 2.3 - 2.5 presents a detailed review of state of the art along with the discussion for gesture recognition, posture recognition and Augmented Reality.

2.1 Fundamental Concepts

In this section, we describe the fundamental concepts and key terminologies which are crucial to build an understanding of the research conducted in this thesis as follows:

- Posture: is defined as a static sign/expression or a hand pose.
- Gesture: is a sequential combination of different instances narrating a particular message (i.e., hand waving).
- Posture vs Gesture Recognition: Posture recognition system mainly relies on the shape, skeleton and structure of the detected hand while

the gesture recognition also depends on detected hand but exploits the temporal information such as trajectories, scales, and orientations.

- **Data Acquisition:** In vision-based hand gesture and posture cues, input sensor (i.e., webcam or Kinect, Asus sensor) gives 2D (i.e., image streams) or 3D data (i.e., image and depth streams), therefore, depth streams can be utilized effectively to extract the region of interest for the underlying scene. For depth streams, we have reviewed 3D approaches along with 2D approach for hand gesture and posture cues.

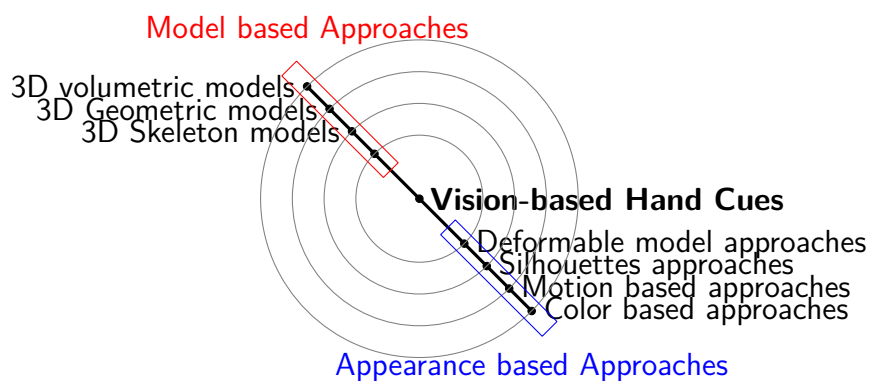


Figure 2.1: Types of vision-based hand cues: model based and appearance based approaches

Considering the contents of this thesis, related work is divided in three main areas namely hand gesture recognition; hand posture recognition and; virtual content augmentation over hand postures. However, in the literature, researchers have used the terms gesture and posture interchangeably (i.e., with and without movement respectively) because of the similar and overlapping variation of algorithms and techniques. Here in this literature review, we have indexed them for hand gesture and postures in a single section as *Vision-based hand cues* but separated them afterwards into hand gesture and posture cues according to the application domains. The reviewed literature is highlighting the proposed approaches along with detailed discussion.

2.2 Vision-based Hand Cues

In the literature, a wide variety of approaches have been adopted to devise methods for vision-based hand gesture and posture cues but here we have

classified them into two main categories namely model based approaches and appearance based approaches as shown in Fig. 2.1.

2.2.1 Model based Approaches

In the model based approaches, 3D hand kinematics (i.e., 3D spatial description of hand) are modelled with certain Degrees Of Freedom (DOF) [24, 25, 26, 27]. The hand parameters are extracted from this model and are matched with already observed images or its features. Keeping in consideration the hand model building process, a large dataset is required with different views and shape information of the hand. In the literature, the model based approaches for hand recognition cues are divided into three main categories namely 3D volumetric models, 3D geometric models, and 3D skeleton models [28, 29]. First, the volumetric models are built based on the hand skeleton and the skin surface rendering information. The examples in this category contain the 3D textured contents, kinematic hand contents and complex 3D hand surfaces. Second, the geometric models are the ones in which different 3D geometrical shapes (i.e., spheres, cylinders, ellipsoids etc.) are utilized to model the hand physical components like fingers, wrist, joints etc. The advantage of this model is that the finger limbs can be modelled using cylindrical parameters like height, texture and radius. Moreover, in these geometrical models, the association of hand physical components is found along with the associated constraints of hand physical components. This association is a crucial part in 3D geometrical models because different joints combine together and thus increase the dimensionality of parameter space. Third, the skeleton model is constructed by exploiting the joint angle parameters together with segmented lengths to generate a skeleton. So, unlike volumetric modeling which deals with all the underlying parameters, the skeleton model generates a reduced set of parameters.

2.2.2 Appearance based Approaches

The appearance based approaches utilizes the visual attributes in image observations. In these approaches, the parameters are not derived from 3D hand description but instead match the target hand appearance to the ones present in the trained dataset (i.e., normally like features). The example set includes the hand features (i.e., contours, edges, image moments, eigenvectors, finger-

tip etc.) which are extracted and compared with the observed features set. However, there are two major issues to be addressed in appearance based approaches namely feature selection and dataset training. Feature selection is an essential step where unique feature representation is indispensable to ensure robustness (i.e., invariance to translation, rotation and scaling) for the classification. Dataset training process ensures that the samples should be sufficient for the optimized learning of classifier.

In the literature, appearance based approaches for hand recognition cues are divided into four main categories namely color based approaches, motion based approaches, silhouette based approaches and deformable model approaches [29]. First, the color based approaches use the special colors for fingers, markers, gloves or special markers to detect the hand [30, 31]. Second, motion based methods utilizes motion to detect the hand features in the image streams [32, 33]. The silhouette based approaches comes next which utilizes the geometric properties of hand such as contours, convexity, bounding boxes, ellipse, rectangularity, centroid and orientation [34]. Fourth, deformable model approaches in which motion and its variants are modelled for active contours and snake algorithms to analyse the hand gesture and posture cues in the underlying scene [35].

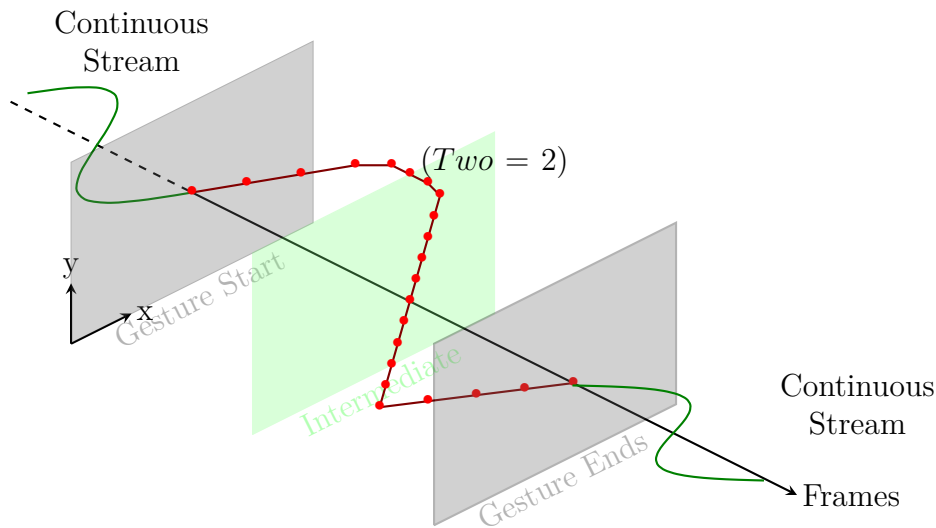


Figure 2.2: Conceptual representation of a gestural action (i.e., digit 2) in a continuous stream.

2.2.3 Discussion

In the above section, main categorization of methods suggested to recognize the vision-based hand cues is presented namely model based and appearance based approaches because of their inherent nature such as, modelling and visual image features respectively. In general, the model based approaches takes the advantage over the appearance based approaches when addressing the problems of self-occluding hand poses. However, this requires intense computational time when the 3D models are constructed containing 3D hand kinematics with certain **DOF**. In contrast, the appearance based approaches rely on the image visual information for hand features extraction which are then passed to the classifier for the recognition. Due to this, the appearance based approaches operate in real-time which motivates the researchers to explore the applicability of appearance based approaches to recognize gesture and posture cues for **HCI**.

2.3 Gesture Recognition

Hand gesture recognition is one of the important research domains of computer vision where the hand detection, feature modelling and classification of patterns (i.e., gestures drawn by hand) are the key components. Briefly speaking, in the gesture recognition, meaningful patterns are identified in the continuous video streams defining the gestural actions as shown in Fig. 2.2. Practically, it is a difficult problem due to the interference caused by hand-hand or hand-face occlusion, lighting changes or background noise causing distortion in the detected patterns and lead to the mis-classification of gestural actions. The researchers have utilized various types of image features such as hand colors, silhouette, motion and its derivatives. However, the literature stated here is focused on appearance based problem and is specifically confined to the color and silhouette-based cues as described in the following.

Bretzner et al. [36] proposed an approach to recognize the hand gestures by utilizing the multi-scale color features. In their approach, the hand shape is detected first which is followed by building a hierarchical model where the shape and color cues are fused at different levels. After that, particle filter is employed for tracking and recognition of the hand state. The drawback of this system is that the static background is used in their approach, so it is

difficult to judge the similar performance with complex and realistic scenes. Similarly, Bellarbi et al. [37] recognized the hand gestures based on color marker detection (i.e., red, blue, yellow and green) attached on two hands. With these mounted color markers, user can draw different gestures such as zoom, move, draw, and write on a virtual keyboard. The basic limitation is that this approach violates the criterion of naturalness due to the use of markers. Chen et al. in [38] proposed a system for hand gesture recognition where the Haar-like features are used to detect the hands. In their approach, instead of using every image pixel, Haar-like computes a rectangular region in the image. Afterwards, a training algorithm based on AdaBoost learning is used to select different Haar-like features for classification in the scene.

Yeo et al. [11] proposed an approach for the hand tracking and gesture recognition in gaming application. In their approach, after the skin segmentation, simple features like position, direction and orientation are measured for the hand gesture recognition and the final result is based on the detected fingertips, to compute the orientation in consecutive frames without any classifier. Afterwards, Kalman filter is employed to estimate the optimal hand positions which results in smoother hand trajectories. The experimental results are reported for the hand gesture dataset which includes open and close palm, claw, pinch and pointing. Similarly, Elmezain et al. [39] presented an approach for the isolated and meaningful gestures utilizing the hand location as its feature using HMM. In their approach, the isolated gestures are tested on the number from 0 to 9 and then combining them to recognize the meaningful gesture based on zero-codeword with constant velocity motion. Finally, the experimental results are reported with the average recognition rate of 98.6% and 94.29% respectively for isolated and meaningful gestures. Huang et al. [40] presents a method for gesture recognition in which they combine spatial and temporal at each frame to extract the feature images. After that, similarity is measured between the extracted features and the Principal Component Analysis (PCA) model (i.e., generated by the training features) using the modified Hausdorff distance algorithm. Finally, the classification is performed using HMM for the set of detected PCA-model (i.e., using similarity measurement) to recognize the gestures. The experimental results present the classification results for 18 different gestures with various shape variations. Li and Greenspan [41] proposed a method for gesture recognition by exploiting the changes in contours. In their approach, the 3D surface is computed from

the 2D radius function and cumulative contour length. Moreover, the classification is performed in two steps where Dynamic Time Warping (DTW) is used to differentiate different gesture models and then matching is performed using Mutual Information. The experiments are conducted on 8 different gestures performed by 5 persons with varied time scales and achieved the recognition rate of 90.0%.

Wen et al. [42] proposed an approach which includes the depth along with the camera information to detect the hand and recognize the gestures. In their approach, first, the skin color is employed along with K-means clustering to segment the hands and is followed by extraction of contour, convex hull and fingertips features to represent the gestures. Similarly, Nanda et al. [43] proposed an approach for hand tracking in cluttered environments using 3D depth data captured from Time of Flight (TOF) sensors. The hand and face are detected by applying the distance transform and k-components based potential fields by calculating weights. The experiments are conducted on dataset comprises of ten people where the hands are tracked under occlusion to classify the signs like step back and stop. However, the suitability of the proposed is effected due to hand shape variation in real-scenarios. In the similar direction, a 3D hand tracking approach is proposed by Argyros et al. [44] where the data is captured by stereoscopic system. The hand blobs are marked by 2D color-based hand trackers in video streams and correspondence is measured by calibration information. Finally, 2D tracking method is applied on hand contours which are aligned in 3D space. The experiments are conducted on various applications such as CD player control by hand gestures. The performance of the system is real-time however the measurement of depth with stereoscopic system is noisy as well as the calibration method is quite complex.

Using Time of Flight (TOF) cameras paired with RGB camera, Van den Bergh et al. [45] proposed a method for gesture recognition. In their approach, the background is first eliminated and hand is detected by applying skin segmentation approach. Further, the average 15 Neighborhood Margin Maximization transformation is measured to build the classifier for gesture recognition, where the Haarlet coefficients are calculated to match hand gestures stored in a database. The experiments are presented for both cameras where the accuracy of 99.2% is acquired with depth-based (i.e., from TOF camera) hand detection whereas accuracy of color-based detection is 92%.

The accuracy of gesture recognition is 99.54% and 99.07% for RGB and depth images respectively. Similarly, Yang et al. [46] recognized gestures for the application of media player. In their approach, eight gestures are recognized based on the hand trajectory features through HMM. Moreover, the tracking of hand is carried out using continuously adaptive Mean-shift algorithm by taking the depth probability and updates its state using depth histogram. The experimental results are presented with the recognition rate of 92%. Raheja et al. [47] proposed a Kinect based recognition system for the hand palm and fingertips tracking in which depth defines the thresholding criteria for hand segmentation. After that, the palm and fingertips are determined using the image differencing methods. The experiment results are presented for the palm center and fingertip tracking with 90% and 100% accuracy respectively. Beh et al. [48] proposed a gesture recognition system by modelling spatio-temporal data in a unit-hypersphere space approach, a Mixture of von Mises-Fisher (MvMF) Probability Density Function (PDF) is incorporated into a HMM. Further, the modeling of trajectories on a unit-hypersphere addresses the constraints of subject's arm length or distance between a subject and camera. The results are presented with public datasets, InteractPlay and University of Central Florida (UCF) Kinect dataset with superior recognition performance compared to relevant state-of-the-art techniques.

2.3.1 Discussion

In the literature, the problem of gesture recognition is addressed using a wide range of approaches for several applications and is impossible to cover it completely in this section. In fact, gesture recognition problem is highly context sensitive such as indoor or outdoor environment, type of gestures (i.e., simple to complex gestural commands), single or multiple interaction mediums and different type of sensors. Due to this, it is difficult to generalize the methodologies suggested for different scenarios like gaming applications, air drawn gestures, and pointing gestures etc. In this thesis, we have investigated in particular the approaches utilizing the appearance characteristics (e.g., color-based and silhouette-based features) and it is observed that a common strategy is adapted by researchers which include hand segmentation, hand tracking, feature modeling and classification. However, the main distinction among the appearance based approaches is determined on the basis of robustness of the

features and the accuracy of classification. For instance, the color-based features are utilized in [36, 37] for the hand gesture recognition which enables robust detection and tracking process but the performance is prominently suffered under varying lightening conditions leading to the significant distortion in the appearance patterns.

The silhouette-based approaches use the hand shape characteristics (i.e., location, direction, orientation, velocity, changes in contours etc.) as features to recognize gestures [38, 39]. Also, in this work [40], edge and image differencing features are fused and then transformed to PCA space for classification. In the same aspect, [41] measured the 3D surface computed from 2D radius function and cumulative contour length for the classification through Dynamic Time Warping (DTW). However, the silhouette-based approaches are more sensitive to noise and have higher computational cost in the pixel-based approaches. To address this, region-based approaches such as Haar like features are employed in [38] which makes them more robust to noise and other lighting variations [49].

Unlike above mentioned approaches which captures through mono-cameras for 2D vision-based approaches, another type of sensor is constructed with the goal of acquiring the depth information. For instance, various researchers employed the range cameras [43] for hand tracking using depth data captured from TOF sensors. This depth information can also be achieved from stereo-cameras (i.e., Bumblebee2 cameras etc.) or through special designed hardware (i.e., sensors built inside Kinect, prime sense or ASUS camera etc.). The advantage of these sensors compared to 2D vision based sensors is the additional depth map which enriches the conventional 2D image streams by 3D information [42, 44, 48]. Similarly, Van den Bergh et al. [45] paired TOF camera with RGB cameras to segment the skin information and hand-face occlusion for hand tracking . In the same context, Yang et al. [46] utilizes the depth probability features inside the adaptive Mean-shift algorithm being continuously updated using depth histogram for the gesture recognition. Similarly, hand palm and fingertip tracking is performed in [47] where the depth defines the criteria of threshold for hand segmentation. Based on these above defined findings, unlike the approaches [38, 39] which are directly utilizing the extracted features at every frame, we have modelled them to extract the robust features before the classification process as presented in Section 4.2.

2.4 Posture Recognition

The hand posture recognition is also one of the important and mature research domains of computer vision where the key elements are the hand detection, posture features extraction and its classification. Briefly speaking, in the posture recognition, the goal is to find the meaningful symbols (i.e., postures) from the continuous video streams. However, it suffers from the varying factors such as lighting conditions, occlusion between hands and face (i.e., hand-hand and hand-face occlusion) which results in the distortion of underlying patterns and cause mis-classifications in posture symbols. In the following, the literature review is focused on the sign language recognition as it is widely used as a body cues for the hearing-impaired people in their communication worldwide. Besides, we have highlighted other related literature for the posture recognition areas where the basic goal is the command and control interfaces.

The sign language recognition is the communication medium for the hearing-impaired people and is divided into three main broad categories namely finger spelling, word-level signs and non-manual signs [50]. In the finger-spelling, sign language alphabets are sequentially presented letter by letter to complete the word. The word-level signs fall in the second category in which the major words communication is carried out whereas in the non-manual features signs, the facial expressions, mouth and body position are also part of the communication. Another application domain is the utilization of detected sign languages symbols to translate between different sign languages, from sign to spoken languages and vice versa (e.g., Zhao et al. [51] proposed a system for the translation from English to American Sign Language). In this related work, the reviewed literature is focused and concentrated on appearance based appearance for the sign language areas (i.e., finger spelling, words and sentence recognition) and command and control interface as follows:

In finger-spelling domain, Freeman and Roth [52] proposed a recognition system based on orientation of histogram for posture (i.e., up, down, right, left and stop) classification. However, the drawback of the system is that for each posture symbol, orientation histograms are trained for each possible angle and therefore, it increases the database training samples size. Handouyahia et al. [53] present a posture recognition system which is based on the shape description using size functions for the International Sign Language (ISL). The size

function is proposed by Frosini [54] for describing and comparing the shapes in which the hand signs are represented using size function with inertia moments containing major and minor inertia axis. The result of this representation is a feature vector which is used to recognize the alphabets in ISL with Neural Networks. The experimental results are based on three different test sets with two subjects each having 10 sequences where the recognition rate in the first and second set are 85% and 89% respectively. In the third test, training samples are divided half from the first and second dataset and achieved the 93% and 96% respectively. Similarly, ElSaadany and Abdelwahab [55] presented a hand posture recognition system to detect the ASL symbols in which first hand segmentation is carried out and for the hand's contours, Histogram of Gradient (HOG) features are calculated where each bin represents a cluster of angles. After that, the PCA is applied on every bin and the classification is performed using Euclidean distance. The experimental results are reported on the Z. Ren's Dataset on 14 ASL signs with 10 persons using leave-one-out strategy and achieved an accuracy of 91.6%. In the similar way, Liwicki and Everingham [56] proposed the finger-spelling British Sign Language (BSL) recognition in which they have used the HOG and classified using HMMs. The experimental results are reported on a dataset of 1000 videos for a single user and achieved recognition rate of 98.9%. Wu and Gao [57] proposed a method based on 3-layered feed forward network to recognize Chinese Sign Language (CSL) alphabets. For the recognition, multi-features and multi-classifiers are proposed which helps in improving the recognition rates for the alphabets. Training of the samples is done by Single Parameter Dynamic Search (SPDS) algorithm on 30 alphabets in CSL which results in learning the parameters for these alphabets. The experimental analysis and results shows that multi-feature and multi-classifier works better in recognition than the single-classifier. The recognition rates for a single classifier for 30 alphabets in CSL are from 80% - 100% while the recognition rates boost up to 96% - 100% in multi-classifier approach.

Isaac and Foo [58] proposed a sign language recognition system for ASL for finger spelling. In their approach, the wavelet features are extracted and neural networks is applied to achieve 99% recognition; however the database size and subjects are not presented. Ayala-Ramirez et al.[59] proposed an ASL recognition system for detecting the hand postures in which the color features are combined with hand geometrical features such as areas, length

and bounding boxes in HCI environment at real-time. The experimental results are presented on six different gestures with the accuracy of 90%. Using Kinect camera (i.e., image along with depth information), Pugeault and Bowden [60] proposed recognition algorithms for ASL finger spelling symbols. In their approach, the finger-spelling alphabets are characterized using shape based approach along with the depth streams which are then classified using random forests. The experimental results are presented on four subjects using their method with and without the fusion of depth streams. Moreover, the application is presented where the singer selects the symbols in case of dubious detected signs.

On the other hand, considering the sign language words and sentences, Braffort [61] proposed an approach for the French Sign Language (FSL) sentences in which the signs are separated into conventional signs, non-conventional signs and variable signs. In their approach, HMM is used for the recognition of 44 FSL sentences. Zahedi et al. [62] proposed a technique for the communication and recognition of word level sign and non-manual features, through appearance based features for ASL. In their approach, features are computed from skin intensity threshold and its derivatives and classified using HMM for words. Similarly, Souza and Pizzolato [63] proposed a system for the sign language words (i.e., hand actions and face expressions) for Brazilian Sign Language (BRSL). In their approach, two-layered structure is proposed where in the first layer, hand shapes are detected using Viola and Jones [64] and Camshift [65] tracker with a Dynamic Virtual Wall Algorithm. In the second layer, the temporal and facial features are measured along with the classification of static gestures to form the words. The experiments results are conducted for 21 subjects using Kinect camera using depth streams with varying kernel functions for the SVM configuration. In the similar way, for the Greek Sign Language (GSL), Vassilia et al. [66] proposed an approach to recognize the isolated and continuous sentences. In their approach, the geometric properties of the hand are utilized to calculate the features for GSL alphabets which are then classified using HMM. The experimental results are presented on different cases with 95.8% and 90.5% for the isolated and sequences of letters respectively whereas achieved recognition rates of 97.4% and 86.2% for the isolated and connected word recognition respectively. Yang et al. [67] present an ASL word recognition system using a time-delay neural network. In their approach, the skin segmentation is carried out along with the motion

segmentation to identify the hands and face. The experimental results present 40 ASL words with the comparison of one versus all trajectories and achieved the recognition rate of 96.21% and 99.02% respectively.

Hand postures are also used in command and control interface as well as in other applications. In this regard, Ren et. al [68] suggested an approach for hand posture recognition using a part based modelling technique. In their approach, earth moving distance is computed from the detected fingers to measure the dissimilarity among the hand shape symbols to recognize the hand postures. The experiments are conducted for 10 postures (i.e., numbers from 1 to 10) with 10 subjects and achieved the recognition rate of 93.2%. However, the results under self-occlusion or occlusion with other hand are not presented. Similarly, Licsar [69] developed a hand gesture recognition system based on the shape analysis of postures. For the classification of hand shapes, modified Fourier descriptors are used which calculates Fourier coefficients for hand shapes. Finally, Nearest Neighbor is employed for the distance metric of modified Fourier descriptor to recognize the hand shapes. The system is tested for a set of nine postures and the feedback is used to train falsely detected postures. The results show 76% to 86% recognition with unsupervised learning whereas in supervised learning, modification of parameters with feedback resulted in 92% to 95% recognition results. In similar way, Malassiotis and Strintzis [70] proposed a 3D hand posture system using appearance based model where 3D information is used to estimate the hand orientation and PCA is applied for dimensionality reduction. Further, 2D hand silhouettes are extracted by applying skin color segmentation and these silhouettes are used for hand posture classification. These silhouette contours are made translation, rotation and scale-invariant through Elliptic Fourier Descriptors where its coefficients are extracted for classification. The results of hand posture recognition system are good but robustness of the system is not given which undermines this approach.

2.4.1 Discussion

The posture recognition is an active domain due to its significance in applications like sign languages among many others. Similar to gesture recognition, it is highly context sensitive and therefore it is not possible to bring the research in a generalized framework because the researchers have taken into account

fixed assumption and requirements. However, most of the reported work followed a standard work-flow including hand segmentation, feature extraction, tracking and classification where the contribution is marked at several level of the processes.

Practically, in the literature review, several researchers used gloves to address the hand segmentation problem. In these works [57], [61], data is acquired using hand gloves avoiding the segmentation problem and to extract features robustly. In these approaches, naturalness criterion is violated by the use of hand gloves. Similarly, in these approaches [56], [59], [69] the hand posture recognition is performed in static backgrounds setting for good segmentation. Moreover, in this approach [56], [59], the constraints are applied to acquire the information about hand such as special clothing restriction where the user is bound to wear the full arm sleeves. Moreover, in this work [55], the constraints such as user's hand should remain closest to the camera or user should wear black belt at the wrist that helps in separating the hand from the arm region are taken into account for hand segmentation and arm pruning.

Another significant challenge faced in posture recognition is to detect the distinct features robustly. The basic attributes for the distinct features should be the invariance to rotation, scaling and translation [69], [67], [70], [68]. However, the disadvantage of these approaches [53], [52] is that they are not invariant to rotation and so tremendous amount of data is required to train with different orientation for every symbol to handle the rotation invariance. Moreover, in the posture recognition, the requirement of high training data and low memory consumption is addressed by [60], Licsar [69] using the restricted set of hand postures. Similarly, the small training data is used in [62] but it has the disadvantage that the utterances cannot be classified correctly in sign language words (i.e., needs more training data). Moreover, the hand tracking is a difficult problem in computer vision where the self-occlusion and occlusion of the hands and face are the pertinent challenges. However, the results of these approaches didn't address the occlusion problem at all [66], [68],[70].

As pointed out earlier that fixed assumptions are taken into account leading to significant constraints in the sign language applications for example: restriction of static background, user's clothing conditions like long sleeves or specific rules for the posture formation. We argue that, such assumptions violate basic criteria of HCI systems that are naturalness and user flexibility.

The proposed research aims to introduce new methodologies in vision-based HCI system that require no restrictions related to background and the user's clothing. Moreover, the tracking (i.e., in case of occlusion between hand-hand or hand face) and feature invariance to translation, rotation and scaling are not addressed by various researchers making scope of their approach restricted to certain situation. Another important issue that is not addressed by researchers in the posture recognition domain is the need of a criterion or function for features modeling in a way to perform the categorization of symbols before classifying them. In this way, the mis-classification occurred between symbols are reduced when they falls into different groups. Moreover, by doing so, the posture symbols need less training samples as well. By employing these findings, we have proposed an approach for posture recognition for a flexible environment (e.g., complex background, no clothing restriction, no special markers etc.) in real-time as presented in Section 4.4.

2.5 Augmented Reality

Augmented Reality (AR) aims to map the virtual components over the real contents in realistic manner. In the literature, components such as visual, graphics, and avatars are augmented in different applications such as in the navigation, games, face recognition, secure identification and many more [71, 72, 73]. The augmentation is mainly performed in two main ways: marker-based AR and marker-less AR [74, 75]. In the marker-based AR system [76], the visual components are displayed on the fixed geometrical patterns (i.e., fiducials) whereas in the marker-less AR system [77, 78, 79, 80], the virtual components are overlaid on the detected objects in the corresponding environment. The marker-less AR systems use the body parts (such as, eyes, face, hands, torso, arms, legs and fingers) to interact with machines for providing an intuitive interface. However, as these body parts have deformable structures, the main challenge lies in extracting the consistent geometry over which AR virtual components (i.e, 3D structures, models etc.) are stitched during free movements. In marker-less AR systems, the goal is to align virtual components with real world to build the perception that two worlds coexist, which however requires the estimation of intrinsic parameters and camera pose at each time instance. A huge variety of literature has been suggested in the domain of marker-based and marker-less AR system but keeping the focus

of the research, we have confined the proposed related work which have utilized fiducial and hand features as a reference medium for virtual components augmentation.

2.5.1 Augmentation

In the marker-based AR system, two prominent works are reported in ARTag [81] or ARToolkit [82] in which the aim is to detect the fiducials using vision-based approaches. The vision-based approaches are normally divided into feature-based and model-based approaches where in the former, the goal is to find the correspondence between 2D extracted image features and their 3D world coordinate system. Finally, 3D coordinates of the features are back projected into 2D image coordinate to get the camera pose with minimization of difference between the corresponding 2D features. In ARToolkit [82], the four corner points of the fiducials are used to estimate the location of 3D object rendered. However, the fiducial markers can take other shapes like circle or rings as well [83]. The model-based approaches utilize a CAD model or a 2D template for which the object features are tracked. Chun and Lee [75] proposed an approach using the stereo cameras in which the 3D hand position and fingertip direction are measured for the interaction with virtual objects. In their approach, various interactions are supported for the augmented virtual object by changing its color and shape. However, the virtual object's augmentation is carried out on the fiducial using the ARToolkit library [82] on the real rendered scene. The advantage of marker based system is to look for specific fiducial features in image but has the disadvantage that the marker should be present in addition to hand to be detected in the scene. Moreover, the detected object can always interfere and occlude with fiducials and therefore, can result in the distortion of marker features.

On the other side, the marker-less AR systems here extracts the hands components by exploiting different features of the hand (i.e., center, hand palm, fingers detection) for the interaction with virtual objects. Keeping this motivation, in [84, 79], the human bare hands are used as a distinctive pattern instead of a marker for which camera pose is estimated by the detected fingertips and then the virtual objects are augmented on hand coordinates. In their approach, the calibration parameters generated from the ground-truth data is utilized to calculate the coordinate's data of the detected fingertip

and the final 3D pose information (i.e., 6-DOF) is mapped over the hand for application of moving and placing objects. However, the limitation of these approaches is that the inspection of the object is hindered when the fingertips are occluded by themselves when the hand is flipping or moving. Moreover, in these approaches, all the five fingers must be detected for the final camera pose estimation and these fingers should be outstretched to accomplish the pose estimation process. With similar assumption of outstretched hand, a marker-less AR system is presented in [85]. In this approach, the stereo camera system with depth information and features of the segmented hand are detected and through pose estimation, augmentation is performed on the hand palm.

Similarly, using two hands, an augmented reality system is proposed by [86, 87] to manipulate the superimposed visual objects on the bare hand with a single camera. In their approach, the left hand is used as a virtual marker and the right hand is used as an interaction interface. Moreover, in their approach, the vision-based 2D interaction interface is developed with the AR object by utilizing the tracked fingertips and controlling it using the hand commands. Recently, in the work [88], Kinect camera is used to track and recognize the hand gestures on which the 3D models are augmented for virtual assembly in AR applications. In their approach, the hand tracking and gesture recognition system is used to detect the user's interaction to select, manipulate and assemble 3D models in the virtual assembly process.

2.5.2 Discussion

The augmented reality for marker-based and marker-less system lays its foundation upon the detected features. Therefore, a lot of effort has been carried out to detect the robust features which can be distorted due to the indoor or outdoor environment, single or multiple interaction objects, lighting systems and different type of sensors. Moreover, due to the context sensitiveness of AR applications, it is difficult to generalize the methodologies for different scenarios. However, we have investigated existing mature works used in marker-based AR system like ARTag [81] and ARToolkit [82]. With this motivation, in [75] detects the hand and its features and then the interaction is carried out on the virtual objects by changing its color and shape. The marker-based systems are accurate using the fiducials but has the limitation of specific designed markers (i.e., black and white with particular dimensions)

are to be developed to be used in AR applications. Moreover, like in [75], the disadvantage is that the markers should be present in the scene in addition to the detected hands and it should not be occluded with hand as well.

In contrast, the approaches for hand marker-less AR systems detect the hand and its features by estimating the camera pose to augment the virtual objects [84, 79, 85]. The main limitation of these approaches is the specific features (i.e., like all the detected fingertips) to be detected on the hand and the specific outstretched hand pose for the pose estimation [84, 79, 85]. The bare hand satisfies the naturalness criteria but in these approaches, specific hand postures with detection of particular features (i.e., fingertips) restrict their applicability in AR applications. However, [86, 87, 88] utilize two hands to make these models flexible for an augmented reality system where left and right hands are used for different tasks (i.e., left hand as a virtual marker, right hand act as an interaction interface etc.). The limitation of these approaches is the complex nature to model the hands and in these approaches, results in the specific pose being detected and tracked. To conclude, we argue that the systems developed in hand-based AR system are able to robustly detect the hands along with its features in an unconstrained environment (i.e., without using fiducials, long sleeves or specific poses) by satisfying the naturalness and intuitiveness criterion.

2.6 Summary and Conclusion

The aim of this chapter is to present an extensive survey of gesture and posture domain along with a detailed insight of related issues. Firstly, this survey begins with the fundamental concepts required to understand this thesis and is then followed by the categorization of approaches employed for addressing the hand gesture and posture recognition issues. Second, keeping in consideration this categorization, the literature review for hand gesture and posture recognition system is presented highlighting the advantages as well as limitation of the proposed approaches. Third, we have provided a detailed review of literature for pose estimation process by presenting the work of various researchers along with their categorical discussion and analysis in the domain of AR. We believe that, this survey on hand gesture, posture recognition and Augmented Reality gives an insight to the readers in these domains and encourages new research.

Segmentation

In computer vision, the direct processing of video streams is computationally expensive (due to the size of the image), so one of the approach is to extract objects of interest by segmentation. In this chapter, based on the underlined related issues in Section 1.3, we aspire to describe the context description in Section 3.1 which is followed by the suggested approach for skin segmentation in Section 3.2. Afterwards, the blob extraction process is presented in Section 3.3 and its refinement in Section 3.4. Further, experimental results are presented in Section 3.5 along with the analysis which is followed by summary and conclusion in Section 3.6.

3.1 Context Description

The detection and recognition is an important and challenging field in computer vision especially body cues (i.e., human face and hands). During the last two decades, the researchers are continuously devising new approaches to detect face and hands efficiently for the HCI applications. In this domain, the interaction medium and the camera settings are selected mainly based on three main parameters: 1) context and scenario 2) problem definition, and 3) objectives. Consequently, these parameters lead to the development of new

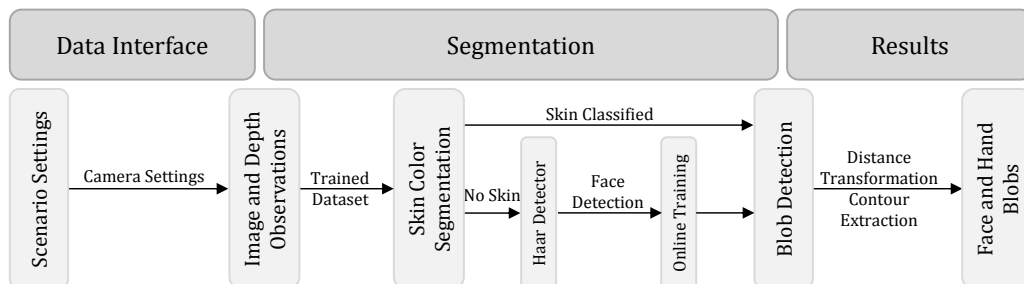


Figure 3.1: Proposed skin segmentation framework.

datasets in image-based HCI for face and hand recognition. For instance, in face recognition, these datasets ranges from face detection of single to multiple subjects along with other parameters including face pose, face location, meaningful and distinctive poses, facial expressions, and different ethnicities under varying conditions such as scene illumination and different backgrounds.

In contrast, the datasets for hand recognition are unconditioned to the presence of face and are highly dependent on the research context and application (i.e., camera can be mounted on the subject’s head or oriented in front of the subject). In similar context, many researchers used gloves or markers for the hand to increase the robustness in recognition process problem which however exclude the segmentation process. But, the bare hands are indeed recommended for HCI application because it satisfies the criteria of naturalness, ease and flexibility [28, 89, 16].

The datasets in the hand classification ranges from detecting either single or both hands in challenging conditions such as occlusions, complex backgrounds, and varying the skin tones due to different ethnicities. In this research, the objectives are defined for two different contexts with respect to camera and its orientation. The contexts are defined as:

3.1.1 Context for Gesture and Posture Scenario

In the first context (**IIKT-GP**), camera is oriented in front of the subject who is communicating with body language cues through hand gestures and postures (see Fig. A.1 in Appendix). The streams of 2D images and depth observations are captured using a Kinect camera [5]. The depth observations are processed to define region of interest (**ROI**) to segment the objects ranging from 30cm to 200cm. Moreover, in experimental scenario settings, upper body structure of the subject (i.e., both hands and face) should be present in the scene to recognize and infer the actions (i.e., subject should be present inside this ROI). The experimental tests in this scenario are conducted for the skin color model with 640×480 pixel resolution under various illumination conditions. In the configuration settings, the default camera parameters such as automatic gain control and white balancing remains the same and are not modified during the entire experimentation process.

3.1.2 Context for Augmented Reality Scenario

The second context (**IIKT-AR**) refers to the scene where camera is adjusted in front of the subject in a tilted manner (i.e., 45° orientation) for AR scenario (see Fig. A.1 in Appendix). In the hand-based AR scenario, augmenting the virtual contents preliminary on hands is the key objective, therefore, only the hands are sufficient to constitute the scene. Besides, in the case of head-mounted camera, capturing face from single camera is not practical. For the hand-based AR application, Kinect camera is not technically feasible because the object of interests (i.e., hands) are in a very close range and therefore, it is very hard to acquire the depth. So, normal webcam is used to capture the image observations directly unlike the earlier mentioned context for gesture and posture recognition. In the configuration settings, 640×480 pixels resolution images are acquired (i.e., both live and recorded) without tuning the camera parameters like automatic gain and white balancing during the entire experimentation process.

3.2 Skin Color Segmentation

Hand and face share a common, unique and quantifiable attribute that is *skin*. The skin information defines itself as an important clue to initiate the segmentation process for transforming the image observations into a calculable form. By this, we mean to separate the interesting and non-interesting pixels in the underlying image. The interesting pixels are accumulated together spatially and constitute blobs (i.e., region of interest) representing the hand and face. The segmentation based on skin color is functioned on the criterion of selecting the portion of image correspond to skin. However, performance of such straightforward segmentation approach is effected due to ambient light, camera outputs and ethnic groups which contain quite varying skin color tune. However, in this research, the proposed approach for segmentation offers both accuracy without sacrificing the generality which is a crucial requirement for HCI applications [89, 28].

Prior to skin color segmentation, we first transform the color representation of captured image observations from the captured image *RGB* to *YC_bC_r* color space. By doing so, skin information is represented by a compact cluster. Technically, in *YC_bC_r* color space, the skin lies in a compact cluster of

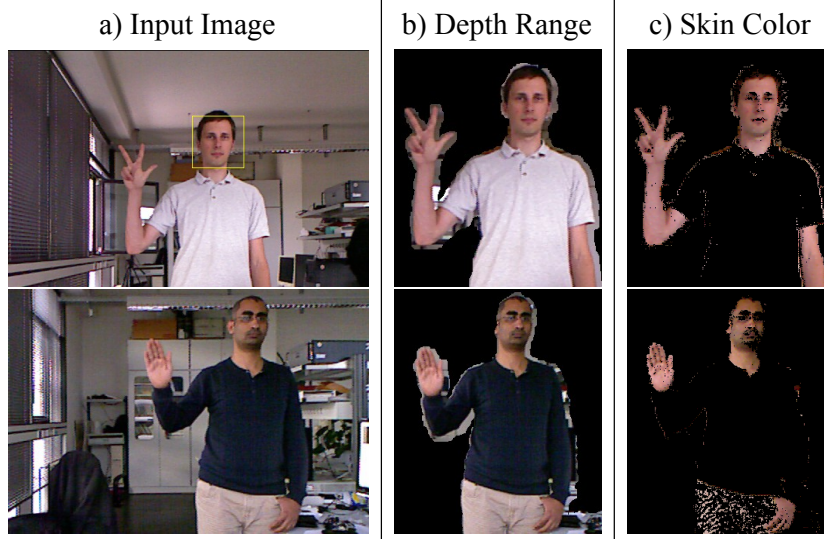


Figure 3.2: Ideal case of segmentation where normal Gaussian distribution works with the trained model having $\mu [Cb Cr] = [97.2 \ 164.4]$ and covariance $\Sigma = [241.57 \ -115.44; -115.44 \ 208.66]$.

chrominance components whereas the effect of brightness variation is reduced by ignoring the luminance (Y) channel. Further, the segmentation process is started by extracting skin color distribution from compact cluster. The skin color distribution is modelled by normal Gaussian distribution [90], characterized by two main parameters (i.e., mean and covariance). The Gaussian model is trained with the database comprising of human skin and other non-skin objects. Mathematically, Gaussian model parameters [90] such as probability, mean and variance of a pixel \mathbf{x} (i.e., 1D) is formularized as:

$$\mathfrak{P}(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\mathbf{x}-\mu)^2}{2\sigma^2}} \quad (3.1)$$

where μ is the mean and σ is the standard deviation.

In the proposed approach, as the chrominance components (i.e., C_b and C_r) are selected for segmentation process, therefore, the input is a 2D vector (i.e., $x = [Cb \ Cr]^T$). Normal Gaussian distribution [90] probability for an observation \mathbf{x} as 2D is calculated as:

$$\mathfrak{P}(\mathbf{x}) = \frac{1}{2\pi\sqrt{|\Sigma|}} e^{-0.5((\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu))} \quad (3.2)$$



Figure 3.3: Original images and skin color segmentation on long and short sleeve images.

The covariance matrix Σ is calculated as:

$$\Sigma = \frac{1}{(n-1)} \sum_{i=1}^n (\mathbf{x}_i - \mu)^T (\mathbf{x}_i - \mu) \quad (3.3)$$

where μ and Σ represents mean vector and covariance matrix respectively. The computed probability $\mathfrak{P}(\mathbf{x})$ categorize the pixels as skin and non-skin pixels as shown in Fig. 3.2 c) and Fig. 3.3 b).

3.3 Blob Detection

The segmentation process gives us interesting pixels using skin color representation for which the chain code representation method is applied to extract the contours [91, 92]. These extracted contours are termed as detected blobs and the blobs with very few contour pixels are ignored. Further, as it can be seen in Fig. 3.2 that when the subject is in front of the camera, there are two different categories of skin blobs which are detected in the scene. First category refers to the *face blobs* and the second category is referred as *hand blobs*. It is important to elaborate it here because in the domain of gesture and posture recognition, we are interested in the hand blobs and the face is treated as a secondary background. Primary background refers to the scene without skin blobs. In the following, we describe how the face blobs are separated from the hand blobs and provide the analysis that our approach is adaptable and make use of trained data to use face blobs as a medium to include other ethnicities as well as the handling the cases when the lightening conditions

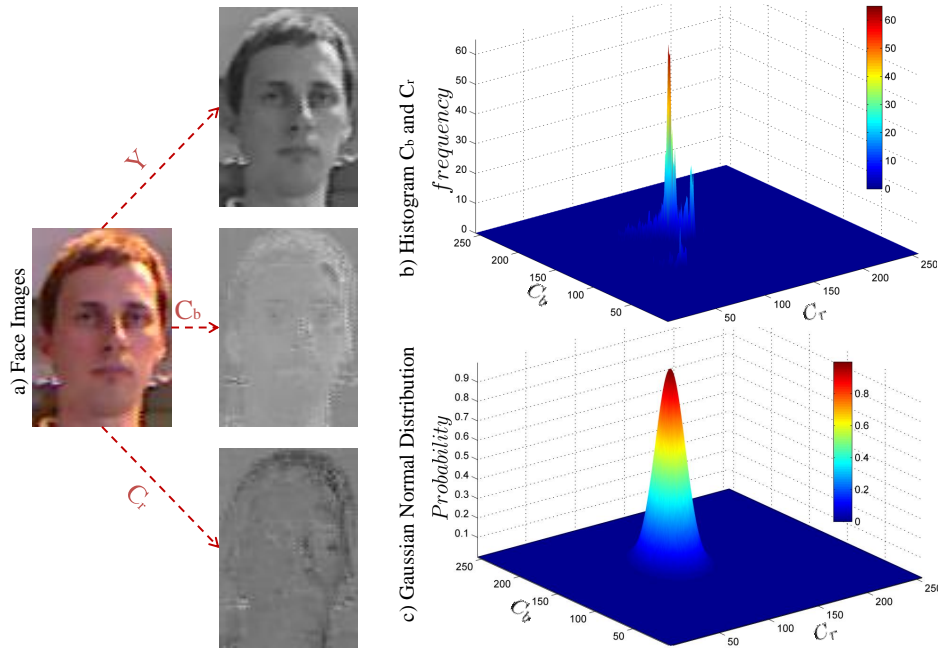


Figure 3.4: a) Face images with Y, C_b and C_r channels. b) Histogram of C_b and C_r channel. c) Gaussian fitted on C_b and C_r channel.

change.

3.3.1 Face Blob

The detected blobs are processed to check whether the face exists in the corresponding blobs or not. For this purpose, Haar-like features [93] is used to detect the face. Once the face is detected, the blob is labeled as secondary background because the main focus of our approach is to recognize the gestures and postures from the input stream (i.e., hand blobs). As the Haar-like features approach for face identification is based on detecting the facial features (i.e., eyes, nose, mouth etc.), so, in the cases when the skin blobs are not detected by trained data, we detect the face and exploit the facial skin information. This facial skin information is used to make an online training of the facial skin pixels using the same procedure described in Section 3.2 and it returns the mean and covariance matrices for the corresponding subjects face as shown in Fig. 3.4. By doing so, we get the flexibility that whenever our skin trained data is unable to identify the skin blobs, the detected face using Haar-like features helps in the hand skin blob detection (i.e., which are

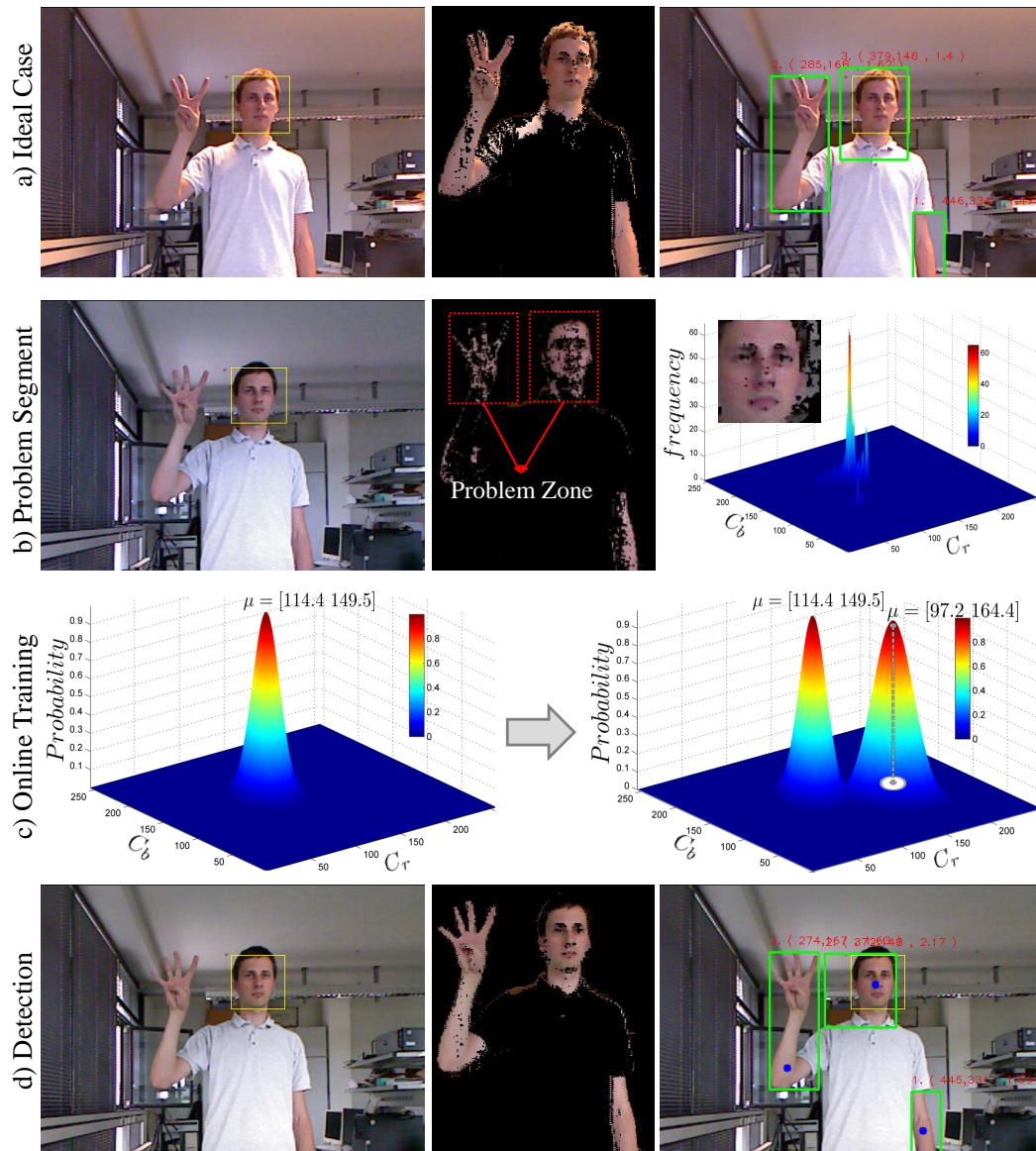


Figure 3.5: Ideal and non-ideal case of skin color segmentation due to the lighting change where normal Gaussian distribution fails to segment the skin, and therefore, the skin is modelled from the detected face samples. Haar-detector is used to detect the face. It results in the mean and covariance matrix of the trained online face samples. (i.e., mean $\mu [Cb Cr] = [123.3 138.802]$ and covariance $\Sigma = [238.1 125.2; 125.2 138.8]$).

considered outliers by the trained skin data). As, the facial skin and hand skin blob has nearly the same color zone, therefore, we can use this skin in-

formation efficiently from the facial skin regions training and detect the skin blobs through the online training data.

Fig. 3.5 presents a scenario containing detected face and hand blobs where a) represents the situation in which the skin regions are detected in the scene through normal Gaussian distribution and then blobs are detected. This is an ideal situation where both the face and hands are detected. However, due to the lightening changes in the scene, there might be some skin pixels which are not classified correctly. This scenario is presented in Fig. 3.5 b) where due to sudden lightening changes, skin segmentation gets worse and the blobs are no more detected. In this case, we detect the face using Haar-like features and the cropped face is used for the online skin training. From the face, we compute the histogram of the skin pixels which are then modelled using normal Gaussian distribution categorized by the mean and covariance matrices. These matrices are added to the actual trained Gaussian to detect the blobs in the next frames. Fig. 3.5 c) presents the trained Gaussian using face data in the left whereas in the right, both trained Gaussians are presented. Fig. 3.5 d) presents the detection results from the new trained Gaussian where it can be clearly seen that the skin is correctly classified and therefore, both the hand and face blobs are correctly detected. In the similar way, we are able to detect the blobs of the subjects having different ethnicities in the corresponding scene (i.e., where the training data is not available or brightness changes rapidly in the scene).

3.3.2 Hand Blobs

After the face detection, hand blobs are left in the image. These blobs vary in number and can be one or two hands in the scene. Moreover, (HCI) interface should consider the basic paradigms such as naturalness and user convenience. For instance, the users should not be restricted to wear short or long sleeves, should not be restricted to wear hand gloves, or to wear any markers on the fingers for hand/fingers identification [30, 31, 57, 61]. Particularly, when the user is wearing short sleeves, segmentation results in the detection of whole detected blob as a hand but actually, it also includes the arm region as shown in Fig. 3.2 and Fig. 3.3. The normal segmentation process does not distinguish the arm region from actual hand region because of the similar color. Many researchers [56, 59] restrict the subjects to wear long-sleeve shirts

for resolving this problem. We have addressed this problem using the distance descriptor in Section 3.4 which detects and eliminates the arm region. Our proposed system fulfils the above mentioned criteria by taking bare hand of the user without any gloves or markers or with long sleeves. In the following section, thus refinement process is presented for the two scenarios described in Section 3.1 on our dataset.

3.4 Refinement using Distance Transformation Descriptor

Distance transformation [94] is an approach which implies on binary images and specifies the distance from each pixel to nearest non-zero pixel. Basically, it is a geometrical operator with huge applicability in computer vision, shape analysis, shape recognition and pattern recognition. Practically, distance transformation has been utilized in literature for the comparison of binary images resulted from the local feature detectors (i.e., corner detectors, edge operators etc.). For example, comparison of binary images through Hausdorff matching approaches is carried out by [95] whereas [96] performs it using Chamfer distance. With the same motivation, distance transformation is utilized for the skeletonization of various shapes in [97]. In this thesis, we take the motivation from these presented applications, and use the distance transformation to extract the hand from blobs consisting of hand and arm and we refer it as *Refinement of skin segmentation*. The motivation of utilizing the distance transformation is to refine and enable the intended application more natural and less-restricted for the users, for instance, the assumptions for short sleeves dress codes.

In the proposed approach, distance transformation computes the distance of each point of plane to the given subset. So, for the binary image $I : \varphi \subset \mathbb{Z}^2 \rightarrow \{0, 255\}$ where the image is interpreted as 2D integer lattice \mathbb{Z}^2 consisting of 0 and 255 values [98], the domain φ is convex and, $\varphi = \{1, \dots, m\} \times \{1, \dots, n\}$, m and n are the number of rows and columns respectively in binary image. In this image, black pixels are represented by 0 where as 255 represents the white pixels. In our case, if the skin region is

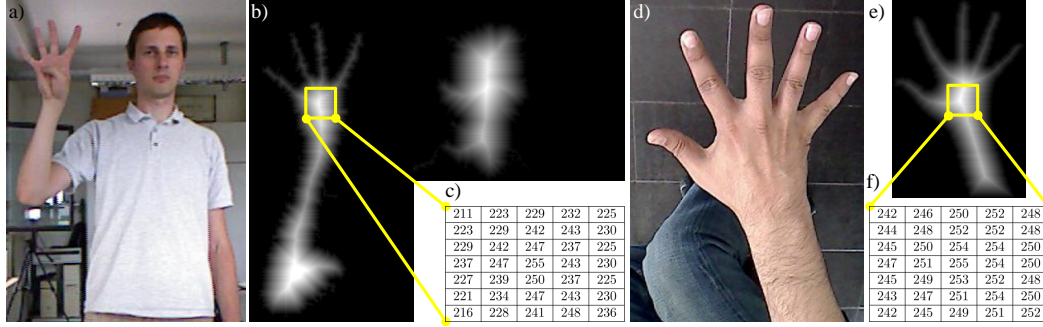


Figure 3.6: a-c) The first scenario a) Original image (Gesture and Posture) b) Distance transformation of the image c) Sample values of distance transformation to extract the palm's center. d-f) The second scenario (Augmented Reality) d) Original image e) Distance transformation of the image f) Sample values of the distance transformation to extract the palm's center.

represented with all the white pixels, we define this region R as:

$$R = \{p \in \varphi | I(p) = 255\} \quad (3.4)$$

In this way, distance transformation generates a map M whose value at each pixel p is the smallest distance from this pixel to background pixel R^b (i.e., non-skin pixels).

$$D(p) = \min\{d(p, q) | q \in R^b\} = \min\{d(p, q) | I(q) = 0\} \quad (3.5)$$

The resulted image D is the distance map of the original image I . To compute the distance map and resulted image D , numerous distance computation methods are employed in the literature such as Euclidean distance, city-Block distance etc. In the proposed approach, we have employed Euclidean distance between two points p and q as $d(p, q)$. It is computed as:

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (3.6)$$

The reason of not using the non-Euclidean distance is that they get extremely unstable to rotations such as in skeletonization and computing the medial axes transformation. Moreover, the shortest path and the maximum object width computed from the non-Euclidean maps may not correspond to the expected practical meaning [99].

In the literature, [100], [56] addressed the gesture and posture recognition problems with the assumption of user's wearing the long sleeves. But in the real scenarios, this assumption is violated especially in our research scenario. Therefore, we address this problem by measuring the parameters of distance transformation [99] as a descriptor. This descriptor helps in determining the hand and palm center, and therefore, eliminates the arm-region thus relaxing the assumption of subject dress code (i.e., shirt with short sleeves). In the proposed approach, we adapt this concept on the detected skin pixels and compute the Euclidean distance to get the transformed distance map using 3×3 window size (i.e., distance of each image pixel to the closest zero pixel and assigning it a score sc) [99]. By using this descriptor, every skin pixel finds its shortest path to the nearest zero pixel. As the palm's center point is normally the farthest pixel in the image from the zero pixel, so, we label this point as the hand's center point. Fig. 3.6 shows the sample images for both scenarios and are described as follows:

- **Gesture and Posture Scenario:** Gesture / posture recognition where the subject is in front of the camera performing gesture/posture with bare hand and short sleeves. Fig. 3.6 (a-c) presents the gesture and posture recognition image where a) Original image of subject, b) Distance transformation of image, and c) Sample binary values of the distance transformation image where the distance scores are presented with the highest scores to extract hand's center accurately.
- **Context for Augmented Reality Scenario:** In the second image presented for AR scenario in Fig. 3.6 (d-f) where d) Original image, e) Distance transformation of the image, and f) Sample values of the distance transformation to extract the hand's center.

It is noticed that distance transformation using the Euclidean distance helps in computing the hand's center accurately with bare hand of the subject in an unconstrained environment.

3.5 Experimental Results and Analysis

The proposed approach is tested on the real-time scenario taken from IIKT dataset posing unique challenges, for example, skin detection under different

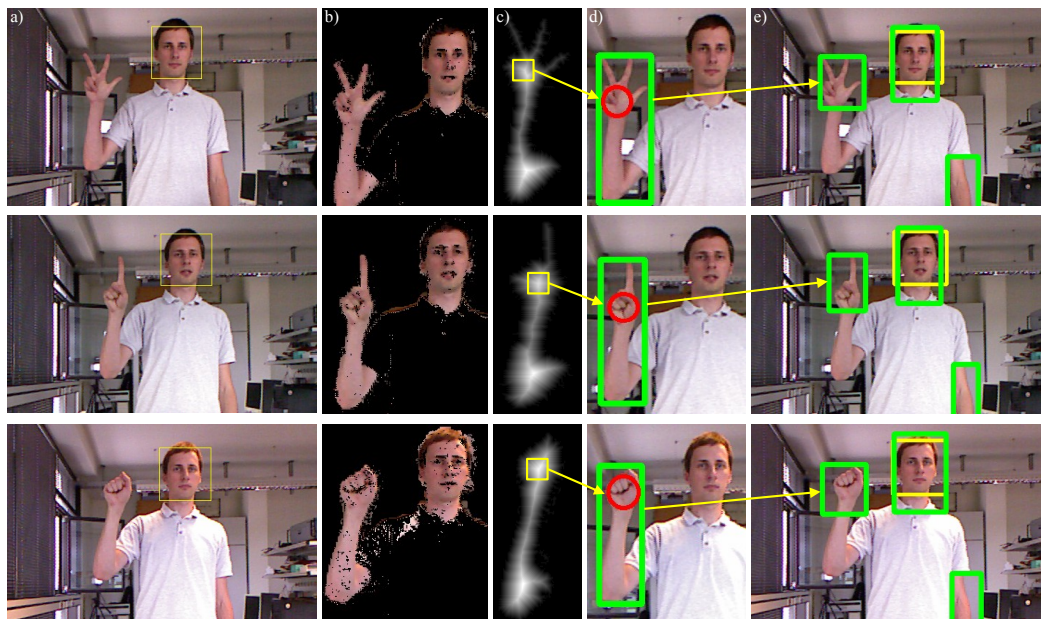


Figure 3.7: Gesture and posture scenario a) Original images b) Skin segmented regions utilizing the depth information c) Distance transformation of the detected blob (i.e., including hands and arm. d) Hand-arm blob extraction from the classified segmented skin pixels (i.e., shown by green rectangle) whereas the results from the distance transformation (i.e., hand center) is drawn as the red circle. e) Hand detection utilizing the distance transformation as well as the face detection (i.e., green rectangle shows the detected hand blob whereas face detection from Haar-like features are shown by yellow rectangle).

illumination conditions and ethnic groups, accurate hand detection for the subject with short sleeves. Fig. 3.7 and Fig. 3.8 demonstrate these issues on two scenarios. In the first scenario (see Section 3.5.1), subjects in front of the Kinect camera (i.e., 3D data streams; image and depth) are drawing the hand gestural and postural signs whereas normal webcam is used for skin segmentation and blob detection in the second scenario (see Section 3.5.2) for Augmented Reality.

3.5.1 Gesture and Posture Recognition Scenario

Fig. 3.7 presents the test sequence from **IIKT-GP** dataset in which skin segmentation, distance transformation and blob detection (i.e., face and hand

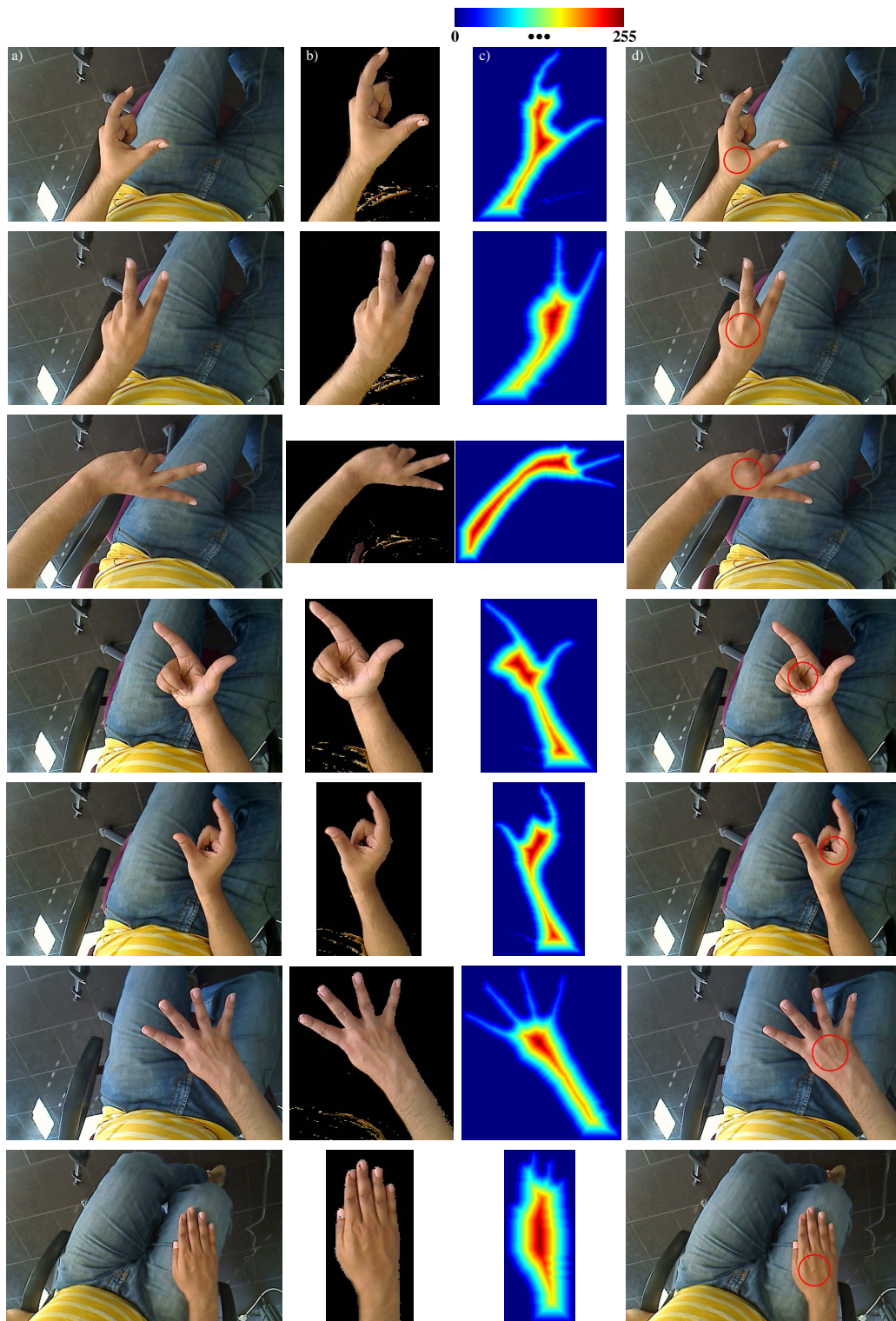


Figure 3.8: Augmented Reality scenario a) Original images b) Skin color segmentation of the images. c) Distance transformation of the images (i.e., red get the higher values 255 and blue gets the minimum value 0). d) Results of distance transformation (i.e., marked as red circle) to extract hand palm's center.

blobs) are performed under different lighting conditions and with short sleeves of a subject. In this sequence, Fig. 3.7 a) presents a scenario where the subject is performing the gestural and postural actions. Fig. 3.7 b) presents the skin segmented regions utilizing the depth (i.e., 30 cm to 200 cm) information. Fig. 3.7 c) shows the distance transformation of the detected blob (i.e., which includes hands and arm. Fig. 3.7 d) presents the hand-arm blob extraction from the classified segmented skin pixels (i.e., shown by green rectangle) whereas the results from the distance transformation (i.e., hand center) are drawn by the red circle. Finally, Fig. 3.7 e) presents the hand detection utilizing the distance transformation as well as face detection (i.e., green rectangle shows the detected hand blob whereas face detection from Haar-like features are shown by yellow rectangle).

3.5.2 Augmented Reality Scenario

Figure 3.8 presents the test sequence from **IIKT-AR** dataset in which the subject is making hand postures for the Augmented Reality application. In this sequence, Fig. 3.8 a) presents original images of the sequence. Fig. 3.8 b) presents the skin color segmentation from normal Gaussian distribution. Fig. 3.8 c) shows the results of the distance transformation (i.e., red get the highest value 255 and blue has the minimum value 0 (see legend in Fig. 3.8)). Fig. 3.8 d) The hand palm is detected through the highest score of distance transformation and is marked with red circle. It can be seen that from the scenarios that the distance transformation is able to detect the hands (i.e., left and right) correctly in this sequence with short sleeves which signifies the performance of the distance transformation approach.

3.5.3 Analysis

In this subsection, the analysis is presented for the refinement process of detected blobs (i.e., consisting of arm and hand) using a distance transformation descriptor. By doing so, the correct detections of hand blobs is carried out for the gesture and posture feature extraction process. Here, in this subsection, we have evaluated the performance of the refinement process by precision and

recall measures as follows:

$$precision = \frac{\text{Correct hand detection (True Positives)}}{\text{Established hand detections (True Positives + False Positives)}} \quad (3.7)$$

$$recall = \frac{\text{Correct hand detections (True Positives)}}{\text{Actual hand detections (True Positives + False Negatives)}} \quad (3.8)$$

where actual hand detections denote the record of ground truth in **IIKT-GP** and **IIKT-AR** datasets. In Table 3.1, based on the computed ground truth and segmentation outcome, the precision and recall are computed to measure the performance of proposed skin segmented approach with and without distance transformation for the detection of hand blobs. It is observed that the results of distance transformation help and improve the blob detection process (i.e., containing hand and arm) to identify the correct hand palm for the feature extraction process. The results show the efficiency of proposed approach and this enables us to detect the hands efficiently under various conditions. Moreover, the difference in the performance is more pronounced with the subject is wearing short sleeves as shown in Fig. 3.7 and Fig. 3.8.

Table 3.1: Precision and Recall: Hand Detection and Palm Center with or without Distance Transformation (DT)

Detection	Precision	Recall
No DT (IIKT-GP)	0.71	0.68
DT (IIKT-GP)	0.95	0.93
No DT (IIKT-AR)	0.72	0.65
DT (IIKT-AR)	0.96	0.95

3.6 Summary and Conclusion

This chapter aims to describe the methodology developed for skin segmentation for two different contexts in video sequences namely the gesture and posture recognition, and Augmented Reality scenario. The significance of skin color segmentation for these contexts is crucial for the higher level processing like feature extraction, hand tracking and classification. Our proposed

method and its modification (i.e., the case where skin segmentation fails and refinement through distance transformation) is geared towards addressing the limitations of existing methods [100], [56], [59], [69] by incorporating the detected face to resolve the skin segmentation issue. Though, segmentation is not the direct contribution of this thesis, but it is an important step to begin with any further steps like tracking, feature extraction and classification. The strength of our proposed approach lies in the fact that when the segmentation process fails, the proposed face detection module is activated and it models the skin patterns from the face to retain the segmentation process. Moreover, pruning of arm by refining the blobs through distance transformation descriptor helps to detect the hand blobs accurately in the image which builds the basis for hand gesture and posture feature extraction process for both scenarios.

Feature Extraction and Tracking

Feature extraction is a crucial step in image processing and computer vision problems. In this thesis, the conducted research is confined to detect and recognize gestural and postural actions, and for this purpose, global and local features are computed from the segmented blobs. Moreover, occlusion is handled through an iterative closest point algorithm which takes local features as observations and resolves the ambiguities between the hands and face to maintain the tracking process. In the following sections, feature extraction process is presented which is used for gesture and posture recognition in Section 4.1 followed by detailed description in Section 4.2, 4.3 and 4.4. The fusion types for different posture features are presented in Section 4.5 which is followed by object tracking (i.e., hands and face) in Section 4.6. Finally, this chapter ends with a summary and conclusion in Section 4.7.

4.1 Features

Features describe an object's underlying characteristics which should be distinctive for every specific action, is a basic requirement of classification. Robust feature detection is a crucial and challenging task under uncontrolled environment; therefore, multiple features are utilized to ensure consistent performance of gesture and posture recognition [16]. Therefore, in the proposed work for posture recognition, different statistical and geometrical features are concatenated to acquire higher performance. The feature set (F_t) for hand gesture and posture recognition system along with fingertip detection at any time instance t is denoted by:

$$F = \{Gstr, Pstr, FT\} \quad (4.1)$$

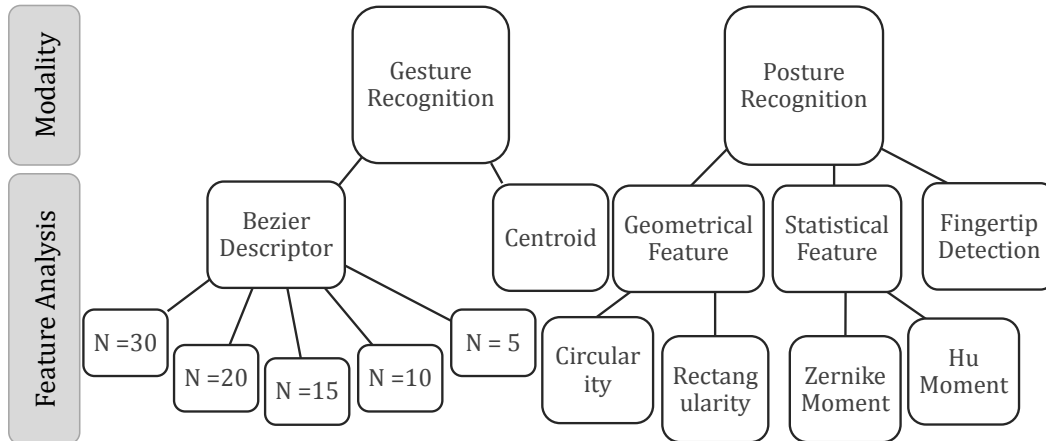


Figure 4.1: Geature and posture recognition: Extraction of gesture and posture features with analysis.

The feature set (F) consists of features for gesture ($Gstr$), posture ($Pstr$) and fingertip (FT) detection as shown in Fig. 4.1. In the following sections, feature extraction process is presented in Section 4.2, 4.3 and 4.4.

4.2 Gesture Features

The gesture process takes input as refined segmented blob (i.e., hand) (i.e., presented in Section 3.4) where the scores from distance transformation help to extract the hand palm as an accurate region of interest to detect hand features. The hand features are inherently static feature set but for the meaningful gestures, these features are measured over a series of time intervals to recognize the complete gestural pattern (i.e., determined over a period of approximately 1 sec) utilizing the Bezier descriptor as described in the following section.

4.2.1 Bezier Descriptors

In the proposed approach, drawn gestures are recognized by modeling the extracted hand centroid points as presented in Section 3.3 (i.e., blob's center from the detected skin segments)¹, collectively using Bezier descriptor as shown in Fig. 4.2 and Fig. 4.3. The proposed approach differs from other

¹The term hand centroid points refers to image-based computed centroid point of the extracted hand. These points act as control points when computing Bezier curves. In the context of Bezier curves, these terms are used interchangeably unless specified.

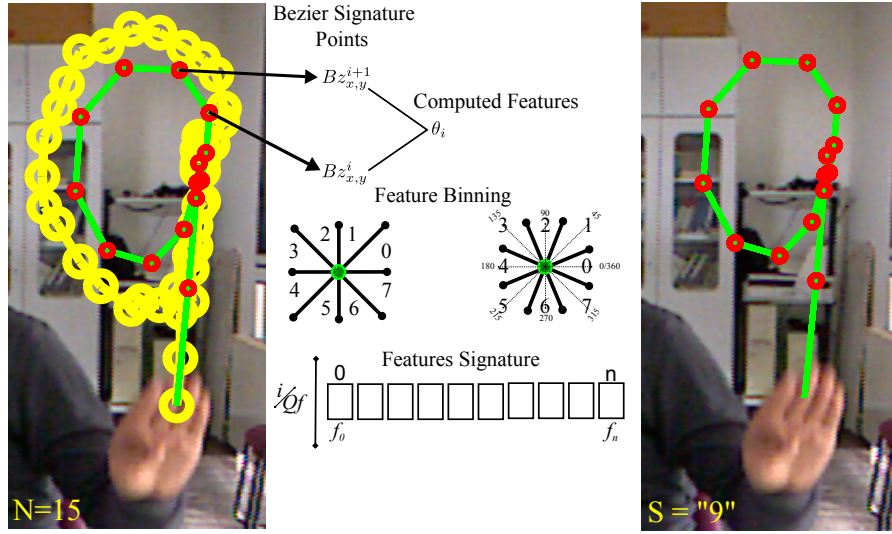


Figure 4.2: Gesture stream with detected hand centroid points (i.e., yellow points) and Bezier points (i.e., red points). The features ϑ are computed from the consecutive Bezier points which are then binned to generate the feature vector signature. The left figure represents the gesture stream (1 second data) by fitting a curve of $N = 15$ Bezier points whereas the right figure shows the fitted Bezier points.

approaches [39, 33] as it is not relying on input hand centroid points but on fitted curves to form smoother trajectories for the classification process. Moreover, the extracted Bezier descriptors produce reliable features even for lower frame rates (i.e., captured frames per second which is not possible in other approaches [39, 55, 33, 11] because of its entire dependence on hand features data). The Bezier descriptor is computed by transforming the hand centroid points (i.e., control points) into a set of Bezier points [101, 102] ($N = 15$) as shown in Fig. 4.3. The Bezier features are computed by finding the difference between two consecutive Bezier points which are then quantized and concatenated resulting in Bezier descriptor (B_d). In the training phase, Bezier descriptor for each gesture symbol is constructed as shown in Fig. 4.6 and Fig. 4.7 which is then used in testing phase to classify the gesture symbols.

Mathematically, Bezier descriptors (B_d) are represented for each gesture symbol through the transformation of control points in the form of Bezier points. In practice, the polynomial or piecewise polynomials are employed to approximate and represent Bezier curves (B). These polynomials [103, 102]

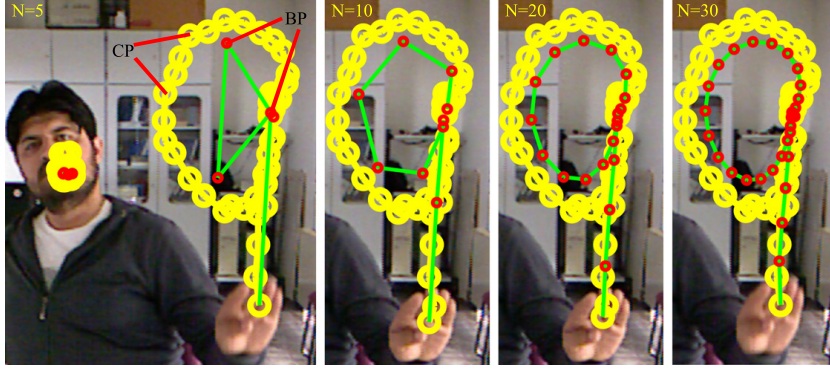


Figure 4.3: Gesture stream with detected hand centroid points CP (i.e., yellow points) and Bezier points BP (i.e., red points). The fitted Bezier points are built from hand control points by varying number of fitting points $N = \{5, 10, 20, 30\}$ which results in trajectories.

can be of various degrees and are defined as:

$$B(t) = \sum_{i=0}^d C_i t^i = C_0 + C_1 t + C_2 t^2 + \dots + C_d t^d \quad (4.2)$$

$$C_i = \frac{n!}{(n-i)!} \sum_{j=0}^i \frac{(-1)^{(i+j)} P_j}{j!(i-j)!} \quad (4.3)$$

The representation of these polynomials results in the approximation of Bezier curves. These Bezier curves can be generalized to higher dimensions which are difficult to control in higher dimensional space, therefore, in the proposed approach, Bezier points are modelled to represent gestures (i.e., 2D space). A linear Bezier curve [104] to represent a line segment has the following form:

$$B(t) = (1-t)P_0 + tP_1; \quad t \in [0, 1] \quad (4.4)$$

where P_0 and P_1 are input points. This mathematical representation [104, 103] is extended to higher dimensional space which has the following form:

$$B(t) = (1-t)^d P_0 + dt(1-t)^{(d-1)} P_1 + \dots + t^d P_d \quad (4.5)$$

$$= \sum_{i=0}^d \frac{d!}{i!(d-i)!} t^i (1-t)^{(d-i)} P_i \quad (4.6)$$

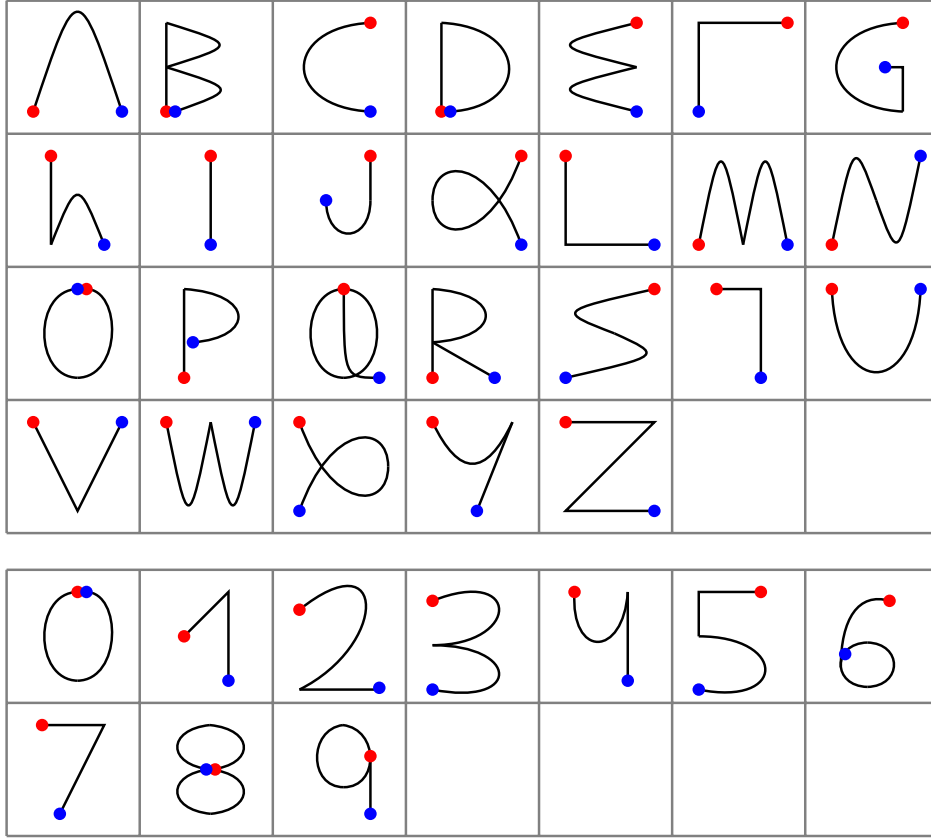


Figure 4.4: The drawing patterns of hand gestural symbols for alphabets and numbers.

$$= \sum_{i=0}^d P_i b_{(i,d)}(t) \quad (4.7)$$

where $b_{(i,d)}$ are Bernstein polynomials which forms the base for all the polynomials of *degree* $\leq d$. The motivation of using Bezier curves is to exploit its convex hull property which ensures that the curve is always confined and controlled with its control points.

Utilizing these Bezier points, orientation ϑ is computed between the two consecutive points to extract the feature vector. The orientation between two consecutive Bezier points $(Bz_{x,y}^i, Bz_{x,y}^{i+1})$ is presented as:

$$\vartheta_i = \arctan \left(\frac{Bz_y^{i+1} - Bz_y^i}{Bz_x^{i+1} - Bz_x^i} \right); i = 1, 2, \dots, T - 1 \quad (4.8)$$

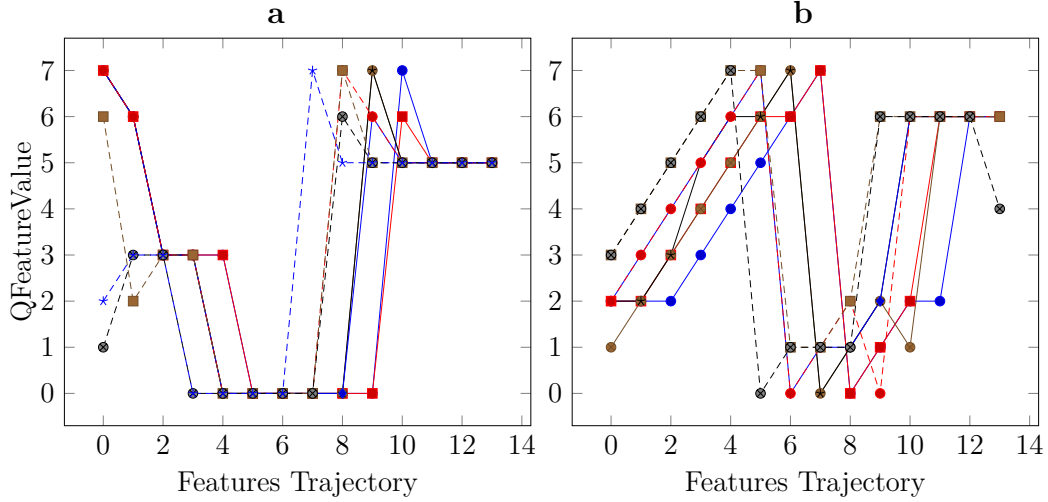


Figure 4.5: Training samples of *Gesture '7'* (left) and *Gesture '9'* (right) used in the classification. X-axis shows the features vector trajectory and the quantized feature values are shown in Y-axis.

where T represents length of gesture drawing path, $Bz_{x,y}^{i+1}$ and $Bz_{x,y}^i$ are two consecutive Bezier points.

4.2.2 Features Binning

In features binning process, the measured orientation ϑ is binned down into different indexes compatible for the classification process. In the proposed approach, orientation ϑ is scaled down into 8 bins with the factor of 45 degrees to get quantized features (Qf) and by concatenating these features; Bezier descriptors ($B_d = \{\vartheta_1, \dots, \vartheta_{T-1}\}$) are formed.

Fig. 4.6 presents the first sequence performed along with its features in graphs. Fig. 4.6 shows the frames of the sequence at various instances with hand centroid points (i.e., control points in yellow color) along with the Bezier points (i.e., red points). The Bezier features (ϑ) are computed from the consecutive Bezier points which are binned to generate the feature vector. Here, the gesture stream (1 second data $\approx 30fps$) at each time instance is represented by 14 Bezier points (i.e., shown by green curve in the tracked trajectory). In the graphs of Fig. 4.6, X-axis shows the feature vector trajectory and quantized feature values are shown in Y-axis. The graph at left shows the quantized feature generated from Bezier points with no gestural pattern

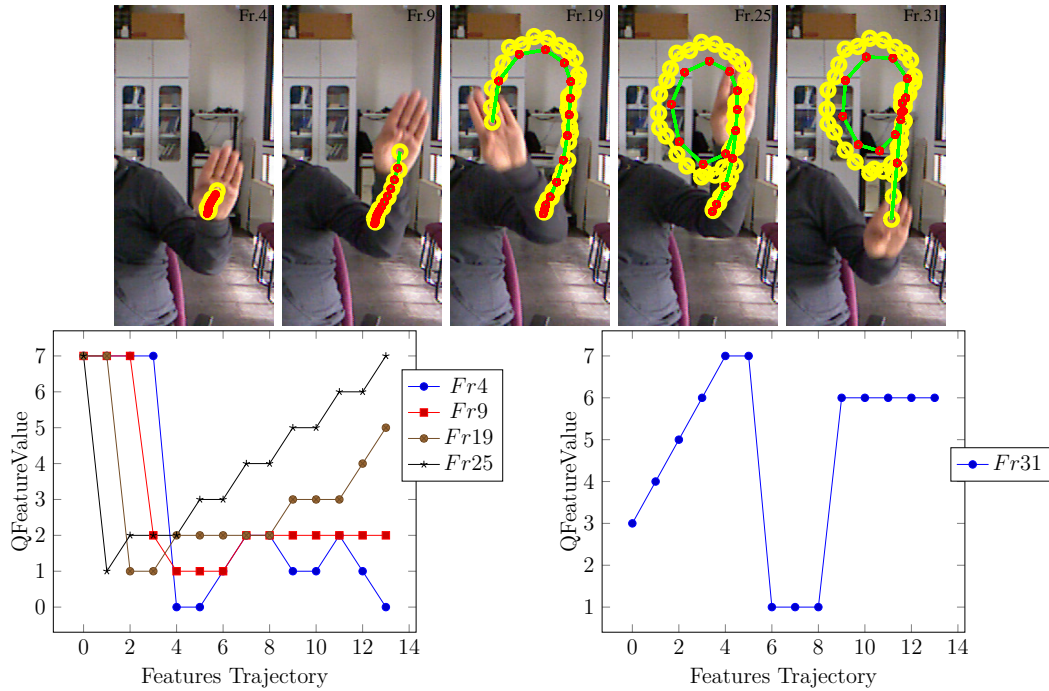


Figure 4.6: Example of gestural action formation for *Gesture '9'* where original detected hand control points (i.e., yellow points) and Bezier signature points (red) are presented. The graphs show the extracted quantized features (Qf) measured from consecutive Bezier points along with the features trajectory. X-axis shows the features vector trajectory and quantized feature values are shown in Y-axis. Left graph shows the features with no gesture detected *Gesture '-1'* (i.e., Fr 4, Fr 9, Fr 19, Fr 25) whereas right graph presents the features for *Gesture '9'* (i.e., Fr 31).

detected at *Fr. 4, Fr. 9, Fr. 19, Fr. 25* whereas the right graph shows Bezier points for detected *Gesture '7'* at *Fr. 31*. Similarly, the second sequence is presented in Fig. 4.7 where the images are shown for *Gesture '7'*. In this sequence, the left graph shows the quantized features from Bezier points where no gesture is detected (i.e., at *Fr. 9, Fr. 16, Fr. 26*) whereas in the right graph, *Gesture '9'* is detected at *Fr. 29* and *Fr. 33*.

4.3 Fingertip Detection

Fingers define the structure and semantics of hand (e.g., hand modeling, ASL, and posture signs). So, the detection of fingers from hand blob is challenging

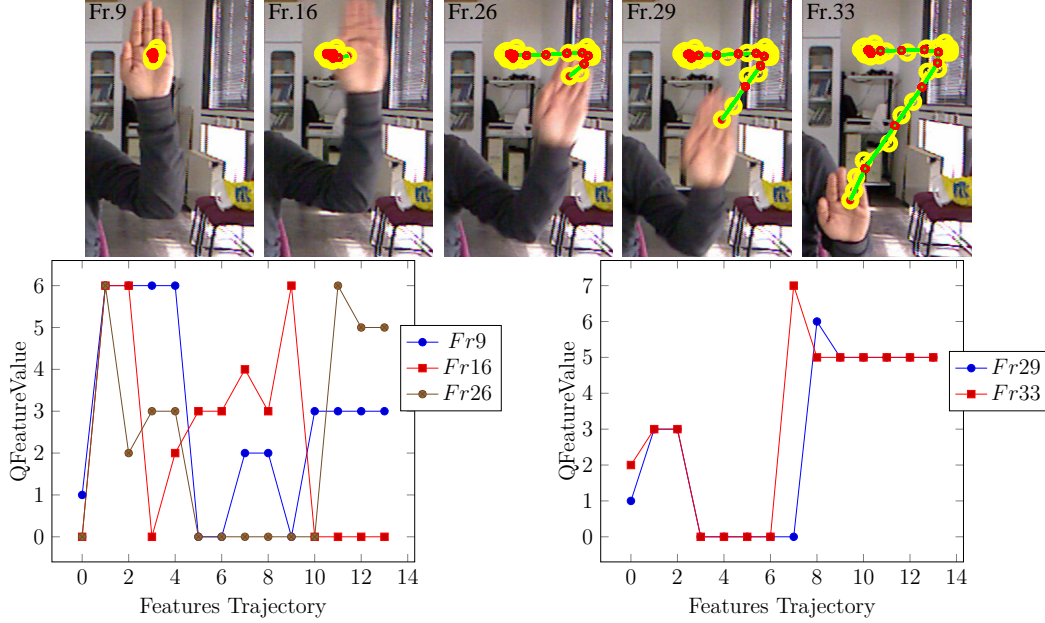


Figure 4.7: Example of gestural action formation for *Gesture '7'*. In the figure, original detected hand control points (i.e., yellow points) and Bezier signature points (i.e., red points) are presented. The graphs present extracted quantized features measured from consecutive Bezier points with features trajectory. X-axis shows the features vector trajectory and the quantized feature (Qf) are shown in Y-axis. Left graph shows the features with no gesture detected *Gesture '-1'* (i.e., Fr 9, Fr 16, Fr 26) whereas right graph presents the features for *Gesture '7'* (i.e., Fr 29, Fr 33).

due to the similar attributes of all the fingers (i.e., same visual characteristics). Based on this fact, the geometrical characteristics (i.e., contour) of the hand are exploited to extract the fingers. Mathematically, the contour segment (C_s) of detected hand in each image I is defined as: $I : C_s = \{P_i\}, i = 1, 2, \dots, N$. P_i are the contour points in the segment with the spatial location (x_i, y_i) . Next, curvature is estimated from the neighbor contour points to detect the fingertip. Mathematically, by utilizing the curvature values, ratio of length (i.e., sum of distances that a curve has) and displacement (i.e., distance measure from the first to last point if curve covers a straight line) are determined. So, for each contour point (P_i), the curvature centered at i for each pixel-wise distance vector (\mathbf{d}_i) is defined as:

$$\mathbf{d}_i = \{d_{(i-M/2)}, d_{(i-(M/2-1))}, \dots, d_i, d_{i+1}, \dots, d_{i+(M/2)-2}, d_{i+(M/2)-1}\} \quad (4.9)$$

$$d_i = \sqrt{(x_i - x_{(i+1)})^2 + (y_i - y_{(i+1)})^2} \quad (4.10)$$

where M is the window size or the number of selected contour points for curvature estimation and is adaptively determined according to the hand palm size. If the palm size is bigger, more contour points are considered in this window for the curvature estimation process and vice versa. Moreover, as each contour point is centered at i , so one half of the window points is selected before and the other half is selected after this point. The computed distances d_i are then summed up to get the final distance s_i for the window as:

$$s_i = \sum_{i-M/2}^{i+M/2-1} \|d_i\| \quad (4.11)$$

The displacement for each contour point P_i is defined as distance of the first and last contour points inside the window:

$$r_i = \|P_{i-M/2} - P_{i+M/2-1}\| \quad (4.12)$$

Curvature κ_i is computed from the following equation:

$$\kappa_i = s_i/r_i \quad (4.13)$$

where i is contour point of the hand at which curvature κ_i is estimated.

The main idea of finding the high curvature values from contour pixels results in the detection of fingertips. In the physical structure of the hand, the fingertips are always present at the high peak points, so we consider only those contour points which lie in these peak regions with normalized distance $nD \geq 0.7$ for fingertips detection. Moreover, an empirical threshold 2.3 is computed in the experimentation process to remove all the points where the curvature is less than this threshold. After pruning these curvature points, only o candidate points are left which are defined as: $L = \{L_1, L_2, \dots, L_o\}$. The next step is to find the corresponding candidate regions from the candidate points L . Therefore, we apply the clustering operation to categorize these candidate points to build candidate regions. A candidate region is the region where the candidate points are spatially clustered together $Q = \{Q_1, Q_2, \dots, Q_u\}$ with u , the total number of candidate regions. Consequently, the number of candidate regions represents the num-

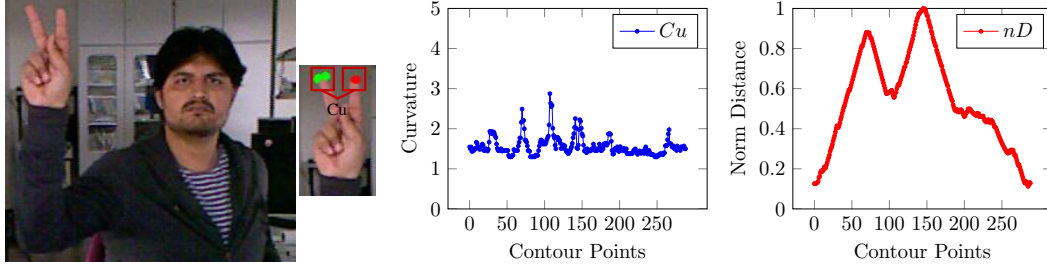


Figure 4.8: Original image with detected fingertips (i.e., Cu (red and green detected points)) in IIKT-GP dataset. The left graph shows hand contour pixels with curvature values whereas the right graph shows the normalized distance (nD) from hand's contour center pixels. Based on our criteria, curvature ($Cu \geq 2.3$) and normalized distance ($nD \geq 0.7$); the fingers candidate points are detected on which the clustering operation is applied to detect two fingertips (i.e., $F = 2$).

ber of fingers (i.e., fingertips of the hand). Finally, the mean of each candidate region points are taken which gives the reference *fingertips candidate points* $F = \mathbf{f}_v, v = \{1, 2, \dots, V\}$ as shown in Fig. 4.8 and Fig. 4.9.

Fig. 4.8 presents the first image and detected fingertips (i.e., Cu). In the graphs, contour pixels of the hand are presented with curvature values in left graph whereas the right graph shows the normalized distance (i.e., nD) from hand's contour center pixels. Moreover, we select the points as a candidate for the fingertip when curvature $Cu \geq 2.3$ and normalized distance $nD \geq 0.7$. In this case, two candidate regions Q_u (i.e., u ; the total number of candidate regions) are extracted which defines number of fingertips detected by taking mean to get the fingertips candidate points (i.e., $F = 2$). Similarly, Fig. 4.9 presents the second image of hand posture and fingertip detection. In this case, two candidate regions are extracted which defines number of detected fingertips.

In this way, the fingertips are detected for at each frame and Table 4.1 presents the confusion matrix of detected fingertips. It is observed that misclassification exists in the neighboring detected fingertips and is due to the reason that only $+/- 1$ detected fingertip is wrong or wrongly detected. A higher rate of mis-classification exists between the hands with two and three detected fingertips. This mis-classification is due to high curvature peaks detected from hand contour points (i.e., noisy contour points) which effect the clustering process and result in the erroneous fingertip detection. In this

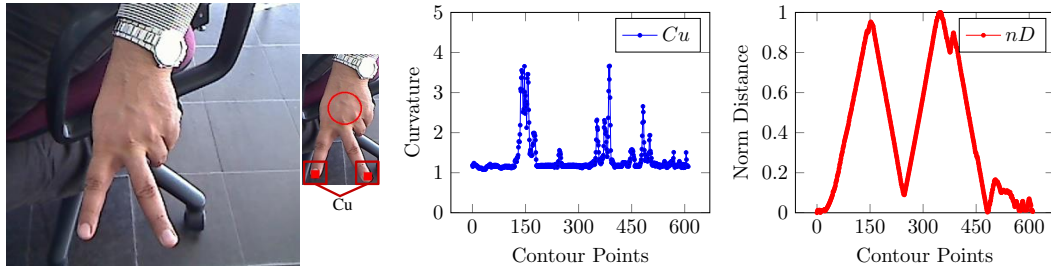


Figure 4.9: Image with detected fingertips (i.e., Cu (red detected rectangles)). The left graph shows hand contour pixels with curvature values whereas the right graph shows the normalized distance (nD) from hand's contour center pixels. Dependent on our criteria, curvature ($Cu \geq 2.3$) and normalized distance ($nD \geq 0.7$); the fingers candidate points are detected on which the clustering operation is applied to detect two fingertips ($F = 2$).

way, the fingertips are efficiently detected for different hand postures using the set criteria by avoiding the wrong detection as shown in Table 4.1.

In this thesis, the outcome of fingertip process is utilized in two approaches namely categorization and hand skeleton as follows:

- **Categorization:** Fingertip detection is utilized to categorize the posture symbols into groups (i.e., criterion based on how many fingers are detected?) as presented in Section 5.2.2 based on the number of participating fingertips. By doing so, the mis-classifications among the hand posture symbols is reduced significantly.
- **Hand Skeleton:** Fingertips are detected as a feature to develop the hand skeleton as presented in Section 7.1 where the fingertips are utilized as a starting path in the path derivation process (i.e., from fingertips to hand palm).

4.4 Posture Features

Fusion of multiple features improves the performance in recognition. With this motivation, in the feature extraction process for posture recognition, statistical and geometrical properties of the hand are computed and integrated resulting in a combined feature vector set denoted as:

$$Pstr_t = \{stat, geo\} \quad (4.14)$$

Table 4.1: Confusion Matrix: Fingertip Detection

		Prediction					
		0	1	2	3	4	5
Truth	FT						
	0	99.8%	0.2%	0%	0%	0%	0%
	1	1.2%	96.8%	2.0%	0%	0%	0%
	2	0%	0%	95.1%	4.9%	0%	0%
	3	0%	0%	0.9%	99.1%	0%	0%
	4	0%	0%	0%	0.6%	99%	0.4%
5	0%	0%	0%	0.6%	1.7%	97.7%	

In the proposed approach, the focus is to compute features (i.e., which are

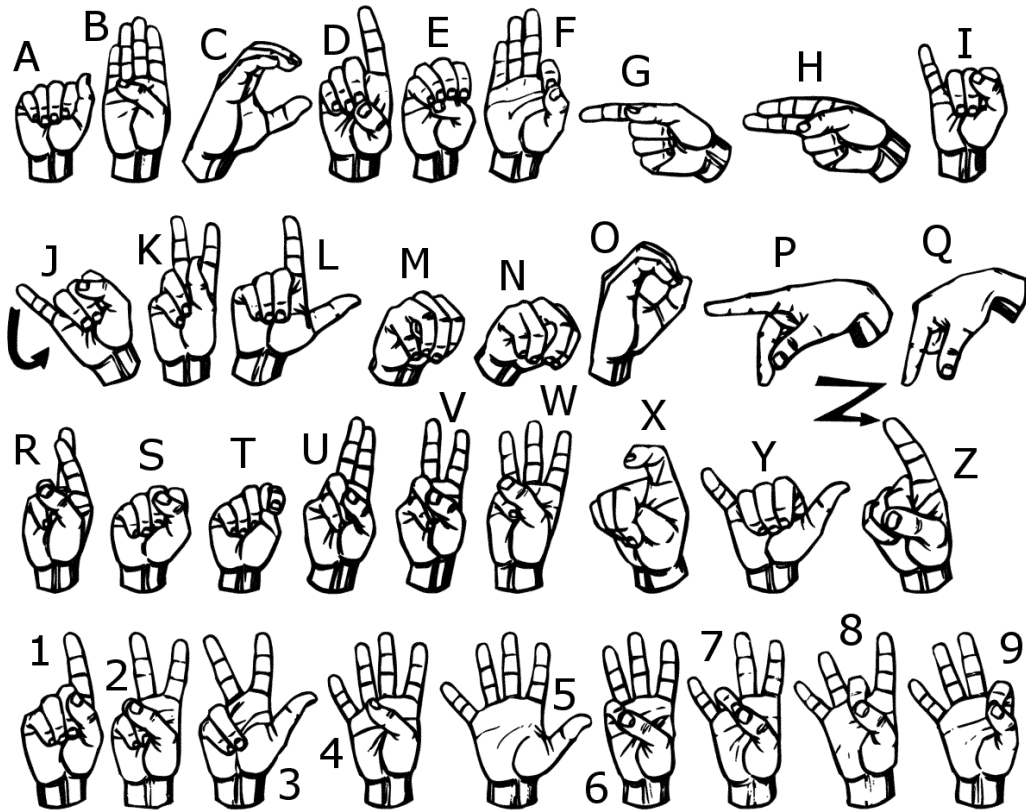


Figure 4.10: The standard ASL finger-spelling alphabets and numbers.

invariant to translation, rotation and scaling) for detecting the American Sign Language (ASL) postures. As, the statistical feature set is derived from moments, therefore, a quick view on the moments is presented before feature derivation process in following section.

4.4.1 Statistical Feature Vectors

In the proposed approach, statistical feature vectors are derived from Hu-Moments [105] containing various properties according to the order of moments. The feature vector of statistical features is:

$$Stat_{pstr} = \{\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \psi_7\} \quad (4.15)$$

Where ψ_1 is the first Hu-Moment and so on.

Hu [105] derived a set of seven moments which are translation, orientation and scale invariant. The equations are computed from the second and third order moments. Hu invariants are extended by Maitra [106] to be invariant under image contrast. Later, Flusser and Suk [107] have derived the moment invariant, that are invariant under general affine transformation. The equations of Hu-Moments are defined as:

$$\psi_1 = \eta_{20} + \eta_{02} \quad (4.16)$$

$$\psi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (4.17)$$

$$\psi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (4.18)$$

$$\psi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (4.19)$$

$$\begin{aligned} \psi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3 \\ & (\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (4.20)$$

$$\begin{aligned} \psi_6 = & (\eta_{20} - \eta_{02}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + \\ & 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \quad (4.21)$$

$$\begin{aligned} \psi_7 = & (3\eta_{12} - \eta_{03})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3 \\ & (\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (4.22)$$

Hu-Moments are derived from a set of seven moments. These seven moments are second and third order moments. The zero-th and first order moments are not used in the feature extraction process. The first six Hu-Moments are invariant to reflection [108] whereas the seventh moment changes the sign. These second and third moments are derived from central moments in which $I(x, y)$ is a digital image with the dimension $M \times N$. The central moments of

order $(p + q)$ is presented as:

$$\mu_{pq} = \sum_y^{M-1} \sum_x^{N-1} (x - \bar{x})^p (y - \bar{y})^q I(x, y) \quad (4.23)$$

To calculate \bar{x} and \bar{y} , the absolute moments of order $(p + q)$ are computed as:

$$m_{pq} = \sum_y^{M-1} \sum_x^{N-1} x^p y^q I(x, y) \quad (4.24)$$

where

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad ; \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (4.25)$$

The central moments up to third order is written as:

$$\mu_{00} = m_{00} \quad (4.26)$$

$$\mu_{01} = \mu_{10} = 0 \quad (4.27)$$

$$\mu_{11} = m_{11} - \bar{x}m_{01} = m_{11} - \bar{y}m_{10} \quad (4.28)$$

$$\mu_{20} = m_{20} - \bar{x}m_{10} \quad (4.29)$$

$$\mu_{02} = m_{02} - \bar{y}m_{01} \quad (4.30)$$

$$\mu_{21} = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2m_{01} \quad (4.31)$$

$$\mu_{12} = m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2m_{10} \quad (4.32)$$

$$\mu_{30} = m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2m_{10} \quad (4.33)$$

$$\mu_{03} = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2m_{01} \quad (4.34)$$

The above moments are translational invariant but to make them scale invariant, these moments are normalized keeping in consideration the order of moment as:

$$\eta_{pq} = \frac{\mu_{pq}}{m_{00}^\gamma} \quad ; \quad \gamma = \left(\frac{(p+q)}{2} + 1 \right) \quad ; \quad p, q \in \{2, 3, \dots, \infty\} \quad (4.35)$$

Properties of Moments:

- *Zero Order Moment:* Zero order moment measures the total mass of the image or any detected object. In the proposed approach, zeroth moment calculates the area of hand (i.e., total number of pixels representing the

hand).

- *First Order Moment:* First order moment values represents the fundamental properties like center of gravity of detected objects (i.e., hands). The center of gravity defines object's location in the image and is used to represent an object in terms of translation invariance in central moment.
- *Second Order Moment:* Second order moment values yield the direction of the main axis of the distribution. Precisely, m_{20} is the variance of the distribution w.r.t X-axis, m_{02} is the variance of the distribution w.r.t Y-axis. m_{11} is the covariance of x and y . The useful features derived from the second order moments are the computation of the principal axes, image ellipse and radii of gyration [109].
- *Third Order Moment:* The third order moment gives image projection on X and Y-axis. The binarized image of hand is the input to determine properties and features of the hand. The third order moments m_{30} , m_{03} define the skewness of image projections. The degree of asymmetry of the distribution is known as skewness. The coefficient of skewness is used to find skewness of the projection. From the analysis of the principal axis, second order moment find the orientation with the help of variance and covariance. However, it does not guarantee a unique orientation because 180 degree ambiguity still exists in it. For this problem, third order moment is used which helps to resolve the ambiguity because 180 degree changes the sign of skewness on either axis.

To summarize, zeroth moment contributes in finding the area of the hand. The first moment determines the mean values of the hand. The second moment contributes to find variance and covariance in both axes. It can also be approximated by an ellipse and represented by the principal axis. Skewness of the hand is computed by third moment. Precisely, all these moments are fused together to give the features set which helps to recognize alphabets and numbers correctly.

4.4.2 Geometrical Feature Vectors

Geometrical feature set contains two features: circularity and rectangularity. These features are computed on refined hand blobs and vary from alphabet

to alphabets. The feature set is described as.

$$Geo_{pstr} = \{Circ, Rect\} \quad (4.36)$$

Circularity: Circularity measures that how much the object's shape is closer to the circle. In the ideal case, circularity is one for circle but circularity ranges from 1 to infinity. As, the values of circularity ranges till infinity, therefore, the normalization step is performed which takes the maximum value from all the posture symbols set. Circularity *Circ* is defined as:

$$Circ = \frac{Perimeter^2}{(4\Pi \times Area)} \quad (4.37)$$

where *Perimeter* is the hand's contour and *Area* is the total number of hand pixels.

Rectangularity: Rectangularity characterizes the similarity of an object with a rectangle. In ideal case, the rectangularity is 1 for the rectangle, however, it ranges from 0.5 to infinity. Similar to circularity, normalization step is also required for the rectangularity to keep it in range (i.e., from 0 to 1). Rectangularity *Rect* is defined as:

$$Rect = \frac{Area}{l \times w} \quad (4.38)$$

where *Area* is the total hand pixels, *l* is length and *w* is width.

4.4.3 Categorization based on Fingertip Detection

The main aim of categorizing fingertip is to divide posture symbols (i.e., ASL alphabets ($Posture Set = \{A, B, C, D, E, F, G, H, I, K, L, O, P, Q, R, U, V, W, X, Y\}$)) into different classes to increase the performance (i.e, recognition rate) [16]. Moreover, it directs the classifier to recognize the symbols within respective groups instead of traversing the whole posture dataset, thus optimizing the classification process along with enhancing the robustness. In Table. 4.2, the posture dataset is divided into four groups depending upon the detected fingertips.

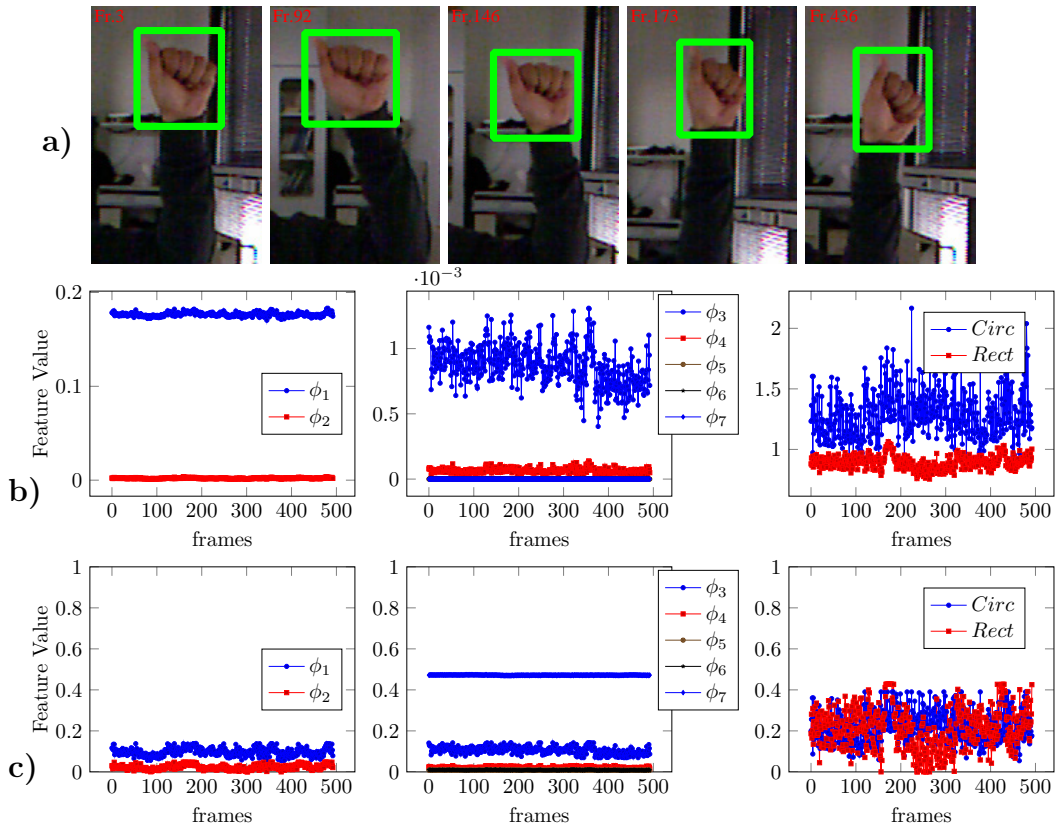


Figure 4.11: a) Sample images from sequence presents the hand posture signs ‘A’ in bounding boxes detected through skin segmented process at various time instances (i.e., *Fr. 3*, *Fr. 92*, *Fr. 146*, *Fr. 173*, *Fr. 436*). b) Upper graphs represent the feature vector set (i.e., left - statistical features second-order moment, middle - statistical features third-order moment and right - geometrical features) c) Lower graphs represent the normalized feature vector set (i.e., left - statistical features second-order moment, middle - statistical features third-order moment and right - geometrical features).

4.4.4 Experimental Results

The experimental setup in posture recognition involves the data acquisition process through Kinect camera, skin segmentation, hand detection and feature extraction. We have demonstrated the applicability of our proposed posture recognition on real-situations with 480×640 pixels image resolution. The experiments are conducted on 1000 video observations of four subjects performing various hand postures (i.e., with varying fingers) with short-to-long sleeves in a flexible manner.

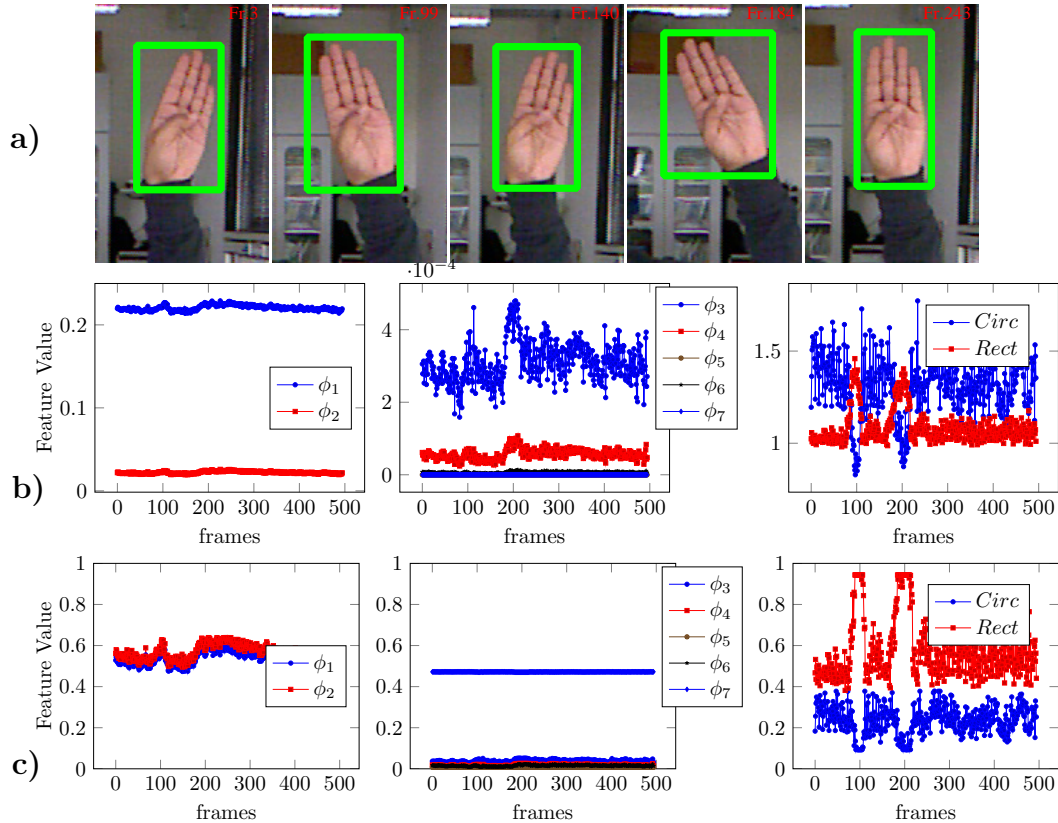








Figure 4.12: a) Sample images from sequence presents the hand posture sign ‘B’ in bounding boxes detected through skin segmented process at various time instances (i.e., *Fr.* 3, *Fr.* 99, *Fr.* 140, *Fr.* 184, *Fr.* 243). b) Upper graphs represent the feature vector set (i.e., left - statistical features second-order moment, middle - statistical features third-order moment and right - geometrical features) c) Lower graphs represent the normalized feature vector set (i.e., left - statistical features second-order moment, middle - statistical features third-order moment and right - geometrical features).

Fig. 4.11 presents the images of first sequence performed by a subject along with its features in graphs. In Fig. 4.11 a) original frames of the stream show the hand posture for sign ‘A’ in bounding boxes detected through skin segmented process at various time instances (i.e., *Fr.* 3, *Fr.* 92, *Fr.* 146, *Fr.* 173, *Fr.* 436). For these hand posture signs, statistical and geometrical feature vector set are computed. Fig. 4.11 b) presents the statistical feature vectors (i.e., features from second-order (left graph) and third-order moments (middle graph)) and geometrical features (right graph). The normalization is

Table 4.2: Fingertip Detection

FT	GroupNr.	Symbols (Alphabets)	Symbols (Numbers)	Example
0	1	A, B, E, F, O, X	0	
1	2	A, B, D, F, G, H, I, R, U	1	
2	3	C, K, L, P, Q, V, Y	2	
3	4	W	3, 6, 7, 8, 9	
4	5	-	4	
5	6	-	5	

performed on the detected feature set to get the normalized feature vector set as shown in Fig. 4.11 c) (i.e., normalized statistical features from second-order (left graph) and normalized statistical third-order moments (middle graph)) and normalized geometrical features (right graph). In these graphs, X-axis shows the frames whereas the feature values are shown in Y-axis. These normalized features are used after the categorization of posture symbols in the classification.

Similarly, Fig. 4.12 presents images of the second sequence performed by a subject along with its features in graphs. Fig. 4.12 a) shows the original frames of hand posture signs ‘B’ at various time instances (i.e., Fr. 3, Fr. 99, Fr. 140, Fr. 184, Fr. 243) with different rotations. For these hand posture signs, statistical and geometrical feature vector sets are computed. Fig. 4.12 b) presents statistical feature vectors (i.e., features from second-order (left graph) and third-order moments (middle graph)) and geometrical features (right graph). The normalization operation is performed on the detected feature set to get the normalized feature vector set as shown in Fig. 4.12 c) (i.e., normalized statistical features from second-order (left graph) and normalized statistical third-order moments (middle graph)) and normalized geometrical features (right graph) which are then used in the classification of posture symbols after the categorization step.

4.5 Feature-Level Fusion for Posture Features

The objective of fusing features is to improve the recognition rate of decision-making process [110]. Normally, there are three types of feature fusion namely

early fusion, cascaded fusion and decision-level fusion.

- **Early Fusion:** In the early fusion, various computed features are integrated into a single feature vector.
- **Cascade-level Fusion:** The cascaded fusion produces the intermediate results by considering each feature at a time and generate the final decision based on the intermediate states.
- **Decision-level Fusion:** In the decision-level fusion, the decision of different features are combined to form a single decision.

In the posture features, the feature-level fusion approach is employed to improve the classification rate in the recognition process. We have fused the statistical and geometrical attributes of the hand to form a feature set for classification process. This feature set is denoted as:

$$Pstr_t = \{\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \psi_7, Circ, Rect\} \quad (4.39)$$

These features are given in Table. 4.3 and their analysis with the classification results are presented in Section 5.2.2.

Table 4.3: Feature Combinations for Posture Analysis

	Hu(2-O)	Hu(3-O)	Geometrical Set	Features Combination ²
A1 (FT)	✓			$\{\psi_1, \psi_2\}$
A2 (FT)		✓		$\{\psi_3, \psi_4, \psi_5, \psi_6, \psi_7\}$
A3 (FT)			✓	$\{Circ, Rect\}$
A4 (FT)	✓	✓		$\{\psi_1, \dots, \psi_7\}$
A5 (FT)	✓	✓	✓	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$

4.6 Tracking

Tracking is an essential and challenging task in computer vision to correctly trace, interpret and infer the underlying object activities in the scene. Tracking maintains the object identities (i.e., by having the position or shape of

²Eq. 4.22 presents the Hu-Moments, Eq. 4.37 presents the circularity feature and Eq. 4.38 presents the rectangularity feature

object) over time in the scene that is used in a wide range of applications in the field of object tracking (such as cars, persons etc.), gesture and posture cues (i.e., hand tracking), eye gaze tracking, face and facial features tracking, event understanding and traffic monitoring etc. The challenge arises in the tracking domain due to abrupt object's motion or camera motion, illumination changes, object's appearance changes, object-object or object-scene occlusion etc.

In the domain of gesture and posture recognition, identities of hands and face should be maintained during the motion as well as when the hands and face occludes each other (i.e., partial and full occlusion). In the literature, various approaches [111, 112, 113] have been proposed to address the tracking paradigms under partial and full occlusions with varying lighting conditions. There are three main categories for the objects tracking namely statistical tracking, kernel tracking and shape tracking.

- In the statistical tracking, state estimation techniques are employed to measure the state of the underlying objects at any time such as Kalman filter, particle filter, and Bayesian inferences [113].
- Kernel tracking employs the template or classifier based approaches to trace the underlying objects in the scene [114].
- Shape tracking utilizes shape or contour matching approaches to track the objects in the underlying scene [115].

In the proposed approach, we have taken shape features and employed (ICP) to align and match the model points (i.e., $2D$ contour points with depth at frame i) to target points (i.e., $2D$ contour points with depth at frame $i+1$). In this step, every model contour point is paired with the closest target contour point and ICP is used to determine the model transformation by minimizing the distance between each pair. The model is then adjusted accordingly and each contour point is paired again with the closest target contour point. ICP is applied to transform the model points to better fit the target contour point and this process is iterated until the paired contour points don't change in ICP alignment. Next, ICP algorithm determines and utilizes the estimated position of detected target points at frame $i+1$ to initialize the model in the next frame $i+2$. Finally, the detected matched points represent a form of motion tracks of the underlying objects (i.e., hands and face). After getting

the tracks of different objects (i.e., hands and face), the association of objects is carried out depending upon the aggregated distances to determine if the track is an optimal one or not (i.e., determine the identities of the objects).

For determining the closest point, we define the set of model points as $P = \{p_1, \dots, p_n\}$ and the target points $Q = \{q_1, \dots, q_n\}$ in vector space X (Euclidean vector space). The ICP algorithm is stated as:

1. Find the closest points Q corresponding to every point from model P using FLANN algorithm [116].
2. Computation of the best rotation (R) and translation (T) matrix between the model points and closest target points.
3. Move the model points to the determined transformation, compute error (i.e., translation error E_t and rotation error E_r).
4. Iterate until it converges completely.

The iteration process ends when termination condition is satisfied which is: if $\|R^i - R^{(i-1)}\| < E_r$ and $\|T^i - T^{(i-1)}\| < E_t$, where E_t and E_r are the thresholds for translation and rotation respectively.

In the proposed approach for gesture recognition, FLANN based matching algorithm is employed to determine the closest points using the randomized kd-trees [117] because of its robust performance in higher dimensional spaces over classical kd-trees. In the classical kd-tree algorithm, data is splitted in half at every level of tree on the dimension where the data has the highest variance. In comparison, the randomized trees are constructed by randomly choosing the split dimension from the first D dimensions where the input data has the highest variance. Moreover, during the tree search, a single priority queue is maintained across all the randomized trees to order the search process by increasing distance to each bin boundary. The result of FLANN matching gives the pairs of $2D$ contour points with depth information from which the transformation between the model and target contour points are determined by taking the mean of these two sets. Afterwards, from the computed distances, correlation matrix is built and Singular Valued Decomposition (SVD) is applied to get the rotation and translation matrices using U and V matrix. Finally, mean square distance between the model and target point sets are compared to recognize whether the two contour point sets are from the same

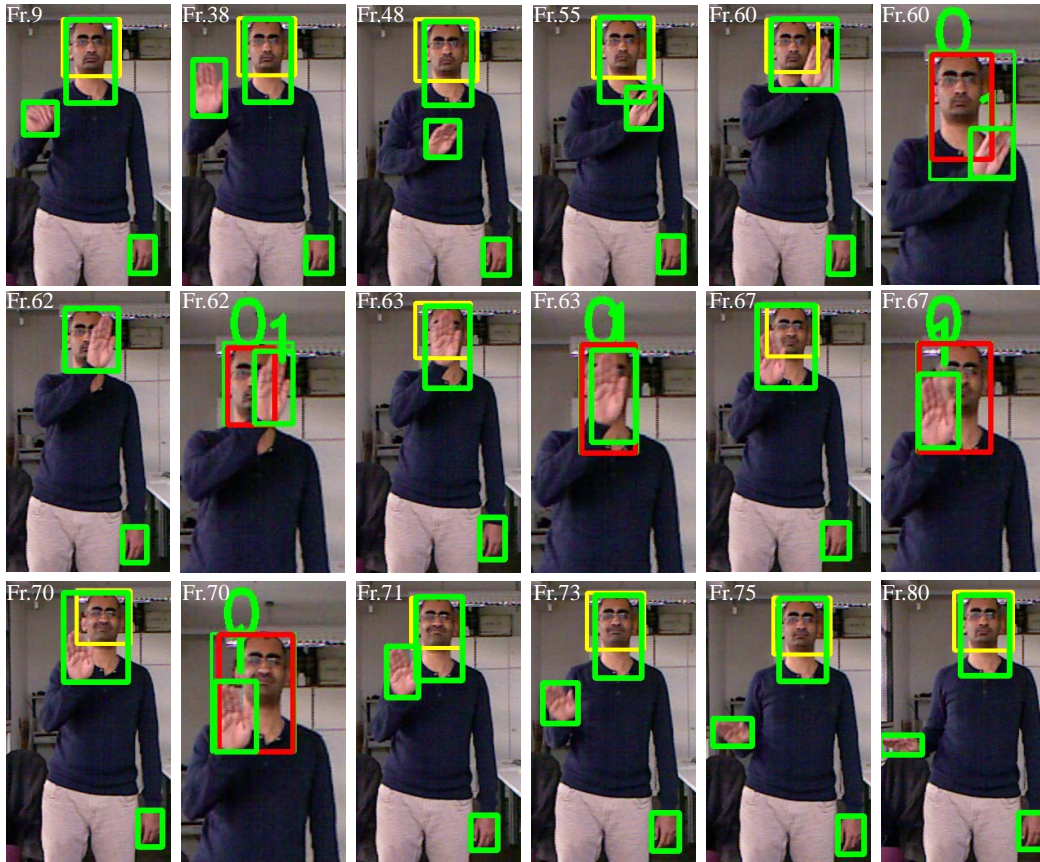


Figure 4.13: Sequence of gestural action where the subject is drawing *Gesture* ‘K’. In the sequence, original detected hand and face blobs are presented (i.e., images with and without hand-face occlusion). It can be seen in the images that occlusion starts from *Fr.* 60 and ends at *Fr.* 70. The identities (i.e., $faceID = 0$ and $handID = 1$ shown only in occlusion) are maintained during the whole sequence utilizing the ICP algorithm with FLANN tracker.

hand or not. If the distance is smaller than a threshold, these two contour points belong to the same hand or face, otherwise these are different hands or face.

Fig. 4.13 presents the gestural action sequence where subject is drawing *Gesture* = *K*. In this sequence, 3D points of extracted blobs (i.e., hands and face) are matched with previous frames 3D points using ICP, equipped with FLANN tracker (i.e., with and without hand-face occlusion). Moreover, it can be seen from the sequence that there is no occlusion between *Fr.* 9 to *Fr.* 59 and ICP with FLANN tracker maintained the hand and face identities

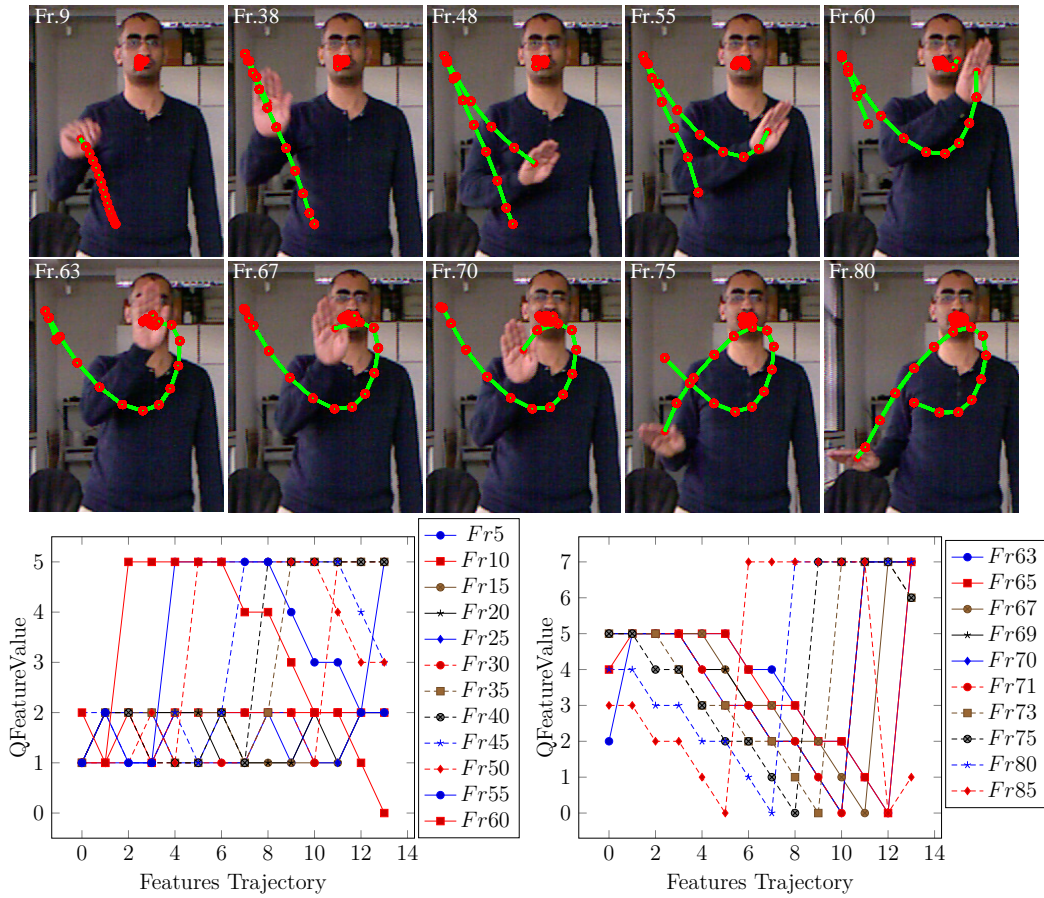


Figure 4.14: Example of gestural action formation where the subject is drawing *Gesture* 'K'. In the sequence, Bezier points (red) are presented. The graphs present extracted quantized features measured from consecutive Bezier points with features trajectory. X-axis shows the features vector trajectory and the quantized features (Qf) are shown in Y-axis. Left graph shows the features with no gesture detected *Gesture* '-1' whereas right graph presents the features for *Gesture* 'K' (i.e., in *Fr.* 70 to *Fr.* 77) along with no gesture detected in the other frames.

as shown in Fig. 4.14 by the Bezier curve trajectories. The hand and face occlusion starts from *Fr.* 60 and ends at *Fr.* 70 in which ICP with FLANN tracker maintained the identities of hands and face (i.e., $faceID = 0$ and $handID = 1$) successfully as shown in the Fig. 4.13 and their trajectories can be seen in Fig. 4.14. After *Fr.* 70, the occlusion process ends and the individual contours are detected for hands and face (see Fig. 4.13 and Fig. 4.14).

Fig. 4.14 presents the Bezier points (i.e., marked with red points) and feature trajectory graphs for the gestural symbol *K*. In graphs, the extracted quantized features calculated from consecutive Bezier points are presented. Left graph shows the features with no gesture detected *Gesture* ‘-1’ whereas right graph presents the features trajectories for *Gesture* ‘K’ (i.e., in *Fr.* 70 to *Fr.* 77) along with no gesture detected in the other frames (i.e., *Fr.* 78 to *Fr.* 85). In this way, we are able to perform gestural and postural actions utilizing the ICP equipped with FLANN tracker under occlusion in complex scenarios.

4.7 Summary and Conclusion

In this chapter, we have presented the feature extraction and tracking approach along with experimental results. The feature extraction starts with the hand gesture features where the Bezier descriptors are constructed from the hand centroid points. Further, the fingertip detection is demonstrated which is used as a criterion to separate the hand posture symbols into groups and thus reduces the mis-classifications among posture symbols. Moreover, the statistical and geometrical features are presented using fusion schemes for the posture recognition. Finally, ICP algorithm is presented and utilized for objects tracking (i.e., hands and face) using FLANN tracker to trace the motion of these objects (i.e., hands and face) spatially over time. Correctly tracking the features over time results in robust identification of objects in the scene.

Classification

Classification is one of the important steps in a computer vision system where the input class is assigned to one of predefined classes. The extracted features are key elements to the classifier because the variability in the features effect the recognition process. In the literature, many classification techniques which includes Gaining Algorithm Learning (**GAL**), K-Nearest Neighbor (**K-NN**), Support Vector Machines (**SVM**), Fuzzy K-Nearest Neighbor (**Fuzzy K-NN**), and Hidden Markov Model (**HMM**) are proposed. In this chapter, **HMM** is presented for gesture classification in Section 5.1 along with experimental results and analysis. Moreover, the posture categorization and classification is presented in Section 5.2 using **SVM**. Finally, this chapter ends with a summary and conclusion in Section 5.3.

5.1 Hidden Markov Models

HMM is a generative classifier and is a mathematical model of stochastic processes where the modelled system is assumed to be Markov process with unobserved hidden states which produces a random chain of outcomes according to the probabilities [118, 119]. In the simple Markov chain process, every state of the model can only observe a single symbol and so, the underlying parameters are the state transition probabilities. However, in the **HMM**, this state is not visible directly, but the output is visible and is dependent on the state. Each state in **HMM** has a probability distribution map over the possible outcomes. So, the sequence of outcomes gives the information about the sequence of states respectively.

5.1.1 Elements of HMM

A Hidden Markov Model [118, 120] is represented by $\lambda = (A, B, \pi)$ and described by the following elements:

1. Set of states $S = \{s_1, s_2, \dots, s_N\}$ where N is the number of states.
2. Set of discrete vector symbols $\varphi = \{\varphi_1, \varphi_2, \dots, \varphi_V\}$ where V is the number of observable symbols at every state.
3. Set of observations (emissions) $O = \{o_1, o_2, \dots, o_T\}$ where T is the length of gesture path.
4. Initial probability for each state π_i , where $i = 1, 2, \dots, N$.

$$\pi_i = P(s_i), \quad \pi_i \geq 0, \quad \sum_i \pi_i = 1 \quad (5.1)$$

5. Transition matrix $A = \{a_{ij}\}$ of size $N \times N$ where a_{ij} is the transition probability from state i to state j at any time instance t :

$$a_{ij} = P(s_t = j \mid s_{t-1} = i), \quad 1 \leq i, j \leq N, \\ a_{ij} \geq 0 \quad \sum_j a_{ij} = 1 \quad (5.2)$$

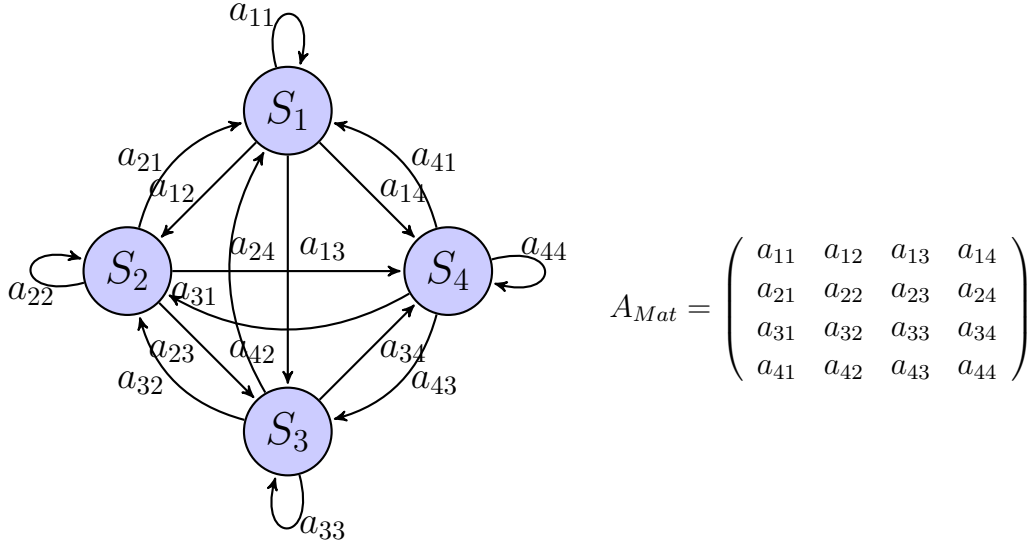


Figure 5.1: The graph presents Ergodic model in which each state can be reached from every other state. A_{Mat} presents the state transition of the Ergodic model.

where a_{ij} is the probability of transition from state s_i at time t to s_j at

time $t + 1$. The sum of entries in each row of matrix A must be equal to 1 because it is the sum of the probabilities of making a transition from a given state to every other state.

6. Observed symbols matrix $B = \{b_{it}\}$ of size $N \times T$ where b_{it} provides us the probability of emitting symbol φ_t at state i :

$$\begin{aligned} b_{it} &= \text{Prob}(\varphi_t | s_i), \quad 1 \leq i \leq N, 1 \leq t \leq T, \\ b_{it} &\geq 0 \quad \sum_t b_{it} = 1 \end{aligned} \quad (5.3)$$

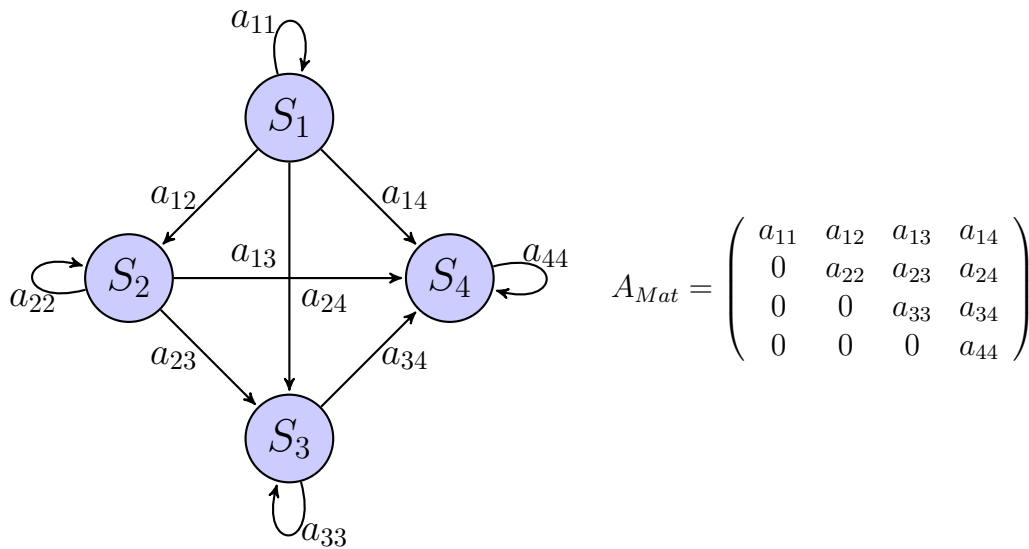


Figure 5.2: The graph presents Left-Right model in which each state of the model can reach itself or to the following state. A_{Mat} presents the state transition of the Left-Right model.

From the above outlined contents, **HMM** has two model parameters (M and N), observation symbols and probabilistic parameters A , B and π . The compact notation of **HMM** can be written as $\lambda = (\pi, A, B)$ where λ is the parameters set of the model.

5.1.2 HMM Topologies

In **HMM**, the topologies are selected by considering the dataset to be trained and recognized which has a significant impact on the recognition process [121].

HMM have three topologies namely Ergodic model (EM), Left-Right model (LRM) and Left-Right Banded model (LRBM). The Ergodic model is a fully connected model where every state can be reached from every other states as shown in Fig. 5.1. The second topology is the Left-Right or Bakis model where each state of the model can reach itself or to the following state [122] as shown in Fig. 5.2. The state indexes in the state sequence remains the same or only increasing whereas the state transitions are not allowed where these indexes are lower than the current state. In the Left-Right model, the state transition has the following form:

$$a_{ij} = 0, \quad j < i \quad a_{NN} = 1, \quad a_{Ni} = 0, \quad i < N \quad (5.4)$$

In this model, the initial state probabilities are written as:

$$\pi_i = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad (5.5)$$

The state sequence starts from s_1 , the transition of states as shown in Fig. 5.2. In the third topology named as Left-Right Banded model (LRBM), the states in this model can reach itself or to the next state. The state transition coefficients in LRBM are presented in Fig. 5.3 and is stated as:

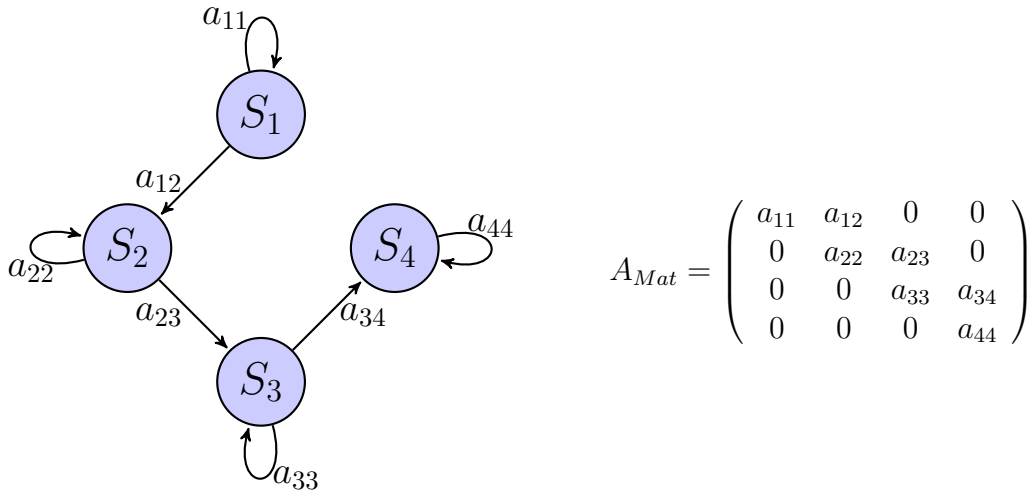


Figure 5.3: The graph presents Left-Right Banded model in which each state of the model can reach itself or to the next state. A_{Mat} presents the state transition of the Left-Right Banded model.

$$a_{NN} = 1, \quad a_{ij} = 0, \quad i < j; \quad j - i \geq 2 \quad (5.6)$$

Looking at these topologies, the underlying topology in **HMM** has to be selected to classify the hand gestures drawn by the subject. In the proposed approach, **HMM** takes the Bezier descriptor as the input derived from the hand gestures temporally and at each time instance. Therefore, we have employed the **LRBM** because this topology is in accordance with the subject drawing gesture at each time instance. In the next section, the classifier **HMM** has to address the following challenges for the training and testing of the gestures.

5.1.3 Problems of HMM

There are three basic problems of **HMM** namely evaluation, decoding and estimation. These problems are presented as follows:

1. **Evaluation Problem:** Given the model parameters and observation sequence O , how to estimate the probability of the observed sequence with the given model parameters $P(O|\lambda)$ (Optimal sequence of hidden states)?
2. **Decoding Problem:** Given the model parameters and observation sequence O , determine the optimal path that best explain these observations O (i.e., with maximum likelihood).
3. **Estimation Problem:** Given the observation sequence O , adjust the model parameters λ to maximize $P(O|\lambda)$.

5.1.4 Solution to Problems

The above stated problems are addressed in this section as:

Evaluation: Given the observation sequence and model parameters, we have to calculate probability of the observation sequence $P(O|\lambda)$ [123]. It is performed through forward algorithm. There are two procedures in this algorithm to calculate the probabilities of forward α and backward β variable respectively. So, to calculate $P(O|\lambda)$, the procedure is to enumerate through every possible state sequence S and measure the corresponding probabilities $P(O|s_1, s_2, \dots, s_t)$ where t is the total number of defined states. The forward

variable [123] $\alpha_t(i)$ defines the probability of the partial observation sequence at state s_i .

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, s_i | \lambda) \quad (5.7)$$

To compute the forward variables at every instance, the following recursive function is used.

$$\alpha_{t+1}(j) = \sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}), \quad 1 \leq j \leq N, \quad 1 \leq t \leq T - 1 \quad (5.8)$$

The initial values of α is computed as follows:

$$\alpha_1(j) = \pi_j \cdot b_j(o_1), \quad 1 \leq j \leq N \quad (5.9)$$

The procedure is terminated when it reaches the length of state sequence T . The final probability $P(O|\lambda)$ is the sum of all the calculated forward variables α as:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (5.10)$$

In the same manner, backward algorithm [123] is proceeded by measuring the backward variables β defined as the probability of partial observation sequences $O = \{o_{t+1}, o_{t+2}, \dots, o_T\}$ at state s_i . The backward variables is computed as:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T, s_i | \lambda) \quad (5.11)$$

Like the forward variables α , the backward variables β are also computed in the similar way using the recursive function as:

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) \cdot a_{ij} \cdot b_j(o_{t+1}), \quad 1 \leq i \leq N, \quad 1 \leq t \leq T - 1 \quad (5.12)$$

where

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (5.13)$$

The final estimation $P(O|\lambda)$ is calculated by multiplying the respective forward α and backward β variables and then summing them up to the number of states N . It is defined as:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_t(i) \cdot \beta_t(i) \quad (5.14)$$

where the multiplication $(\alpha \dots \beta)$ is performed for each state S_i at time instance t results in the respective estimation.

$$\alpha_t(i) \cdot \beta_t(i) = P(O, s_i | \lambda), \quad 1 \leq i \leq N, \quad 1 \leq t \leq T \quad (5.15)$$

As a result, this estimation gives us the probability of observation sequence $P(O | \lambda)$, thus solving the evaluation problem.[123]

Decoding: To find the optimal path state sequence which best defines the observations O with maximum likelihood, Viterbi algorithm is used [124, 125]. Viterbi is a dynamic programming algorithm to find the optimal sequence of hidden states called as Viterbi path and results in the sequence of observed events. Viterbi algorithm is similar to forward variable $\alpha_t(i)$ in evaluation solution but the difference lies in the maximization at each stage from the previous states in the recursion. In the computation step of the Viterbi algorithm [126], an auxiliary variable δ is defined which presents the maximization function and is presented as:

$$\delta_t(j) = \max\{P(o_1, o_2, \dots, o_t, s_1, s_2, \dots, s_t | \lambda)\} \quad (5.16)$$

Taking into account the matrices defined in HMM elements, Viterbi algorithm is presented in four steps namely initialization, recursion, termination and reconstruction [126]. These steps are presented as:

- Initialization: In the initialization step, delta δ and ϕ function are defined as: *for* $1 \leq i \leq N$,
 - a) $\delta_1(i) = \pi_i \cdot b_i(o_1)$
 - b) $\phi_1(i) = 0$
- Recursion: In this step, delta δ and ϕ function are recursively computed taking into account the previous states and is defined as:

for $2 \leq t \leq T, \quad 1 \leq j \leq N$,

 - a) $\delta_t(j) = \max_i[\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(o_t)$
 - b) $\phi_t(j) = \arg \max_i[\delta_{t-1}(i) \cdot a_{ij}]$
- Termination:
 - a) $p^* = \max_i[\delta_T(i)]$
 - b) $q_T^* = \arg \max_i[\delta_T(i)]$

- Reconstruction: for $t = T - 1, T - 2, \dots, 1$

$$q_t^* = \phi_{t+1}(q_{t+1}^*)$$

The output is the optimal state sequence $\{q_1^*, q_2^*, \dots, q_T^*\}$. In optimal sequence, $\phi_t(j)$ represents the index of state j at time t , and p^* is the state optimized likelihood function.

Estimation: In estimation problem, the key issue to address is the adjustment of model parameters λ to maximize $P(O|\lambda)$. It results in the optimal model parameters which best describe the observation sequence O and therefore, gives us the parameters to train the HMM classifier denoted as O_{train} . So, for the training process, Baum-Welch (BW) algorithm is used to optimize the $\lambda = (A, B, \pi)$ with maximum likelihood $P(O|\lambda)$ [127]. Baum Welch (i.e., also called Forward-Backward algorithm) is an expectation maximization algorithm as it is based on forward alpha α and backward beta β variables [128].

Given a set of observation sequences $o_{train} \in O$, BW measures the posterior and maximum likelihood estimation for the HMMs parameters (A, B, π) . In its computation along with forward α and backward variables β , two auxiliary variables are also formed to define the transition probability (i.e., probability of traversing an arc from state i at time t to state j at time $t + 1$) and state probability. Mathematically, the transition probability ξ is presented using the forward and backward variables as:

$$\xi_t(i, j) = \frac{\alpha_t(i) \cdot a_{ij} \cdot \beta_{t+1}(j) \cdot b_j(o_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) \cdot a_{ij} \cdot \beta_{t+1}(j) \cdot b_j(o_{t+1})} \quad (5.17)$$

Similarly, using the forward and backward variables, the state probability (i.e., posterior probability) gets the form as:

$$\gamma_t(i) = \frac{\alpha_t \cdot \beta_{t+1}}{\sum_{i=1}^N \alpha_t \cdot \beta_{t+1}} \quad (5.18)$$

where $\gamma_t(i)$ is the state probability i at time t for given model parameters and observation sequence. The relationship between $\gamma_t(i)$ and $\xi_t(i, j)$ is written as:

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j), \quad 1 \leq i \leq N, \quad 1 \leq t \leq M \quad (5.19)$$

Thus, the Baum-Welch algorithm [127] adjusts the new parameters of the HMM with maximum likelihood of the criterion $P(O|\lambda)$. Given the parameters $\lambda = (A, B, \pi)$, $\hat{\alpha}$ and $\hat{\beta}$ is computed using the recursive equations for α and β . Moreover, the auxiliary variables $\hat{\xi}$ and $\hat{\gamma}$ are calculated by ξ and γ , respectively. The parameters of HMM updated using the following equations:

$$\hat{\pi} = \gamma_1(i), \quad 1 \leq i \leq N, \quad (5.20)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^T \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq N, \quad (5.21)$$

$$\hat{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j) \cdot \omega_{k, o_t}}{\sum_{t=1}^T \gamma_t(j)}, \quad 1 \leq i \leq N, \quad 1 \leq k \leq M, \quad (5.22)$$

where ω_{k, o_t} is defined as follows:

$$\omega_{k, o_t} = \begin{cases} 1 & k = o_t \\ 0 & \text{otherwise} \end{cases} \quad (5.23)$$

5.1.5 Experimental Results

The experimental setup involves the tasks of data acquisition, feature extraction for gesture and classification. Moreover, we have demonstrated the applicability of our proposed system on real situations where the gestures are recognized by satisfying the criteria of ease, flexibility and naturalness. The proposed framework runs with real-time processing at 25 fps on Intel Processor 2.83GHz, 4 cores hardware configuration with 480×640 pixels image resolution. The experiments are conducted on 15 video observations per gesture (i.e., about 50000 samples) of 6 subjects performing various hand gestures wearing short-to-long sleeves and the gesture dataset contains the symbols from $A \rightarrow Z$ and $0 \rightarrow 9$.

Fig. 5.4 presents the sequence where the subject is drawing the *Gesture* ‘8’ along with the feature trajectories in graphs. In this sequence, the features are computed from hand centroid points and transformed to Bezier points $N = 15$ (i.e., marked with red color). The consecutive Bezier points are then utilized to extract Bezier features ϑ and are then binned to build the Bezier descriptor. These Bezier descriptor features are used inside **HMM** to recognize the gestural actions. The left graph shows the quantized Bezier descriptor generated from Bezier points with gestural action *Gesture* ‘8’ detected at Fr

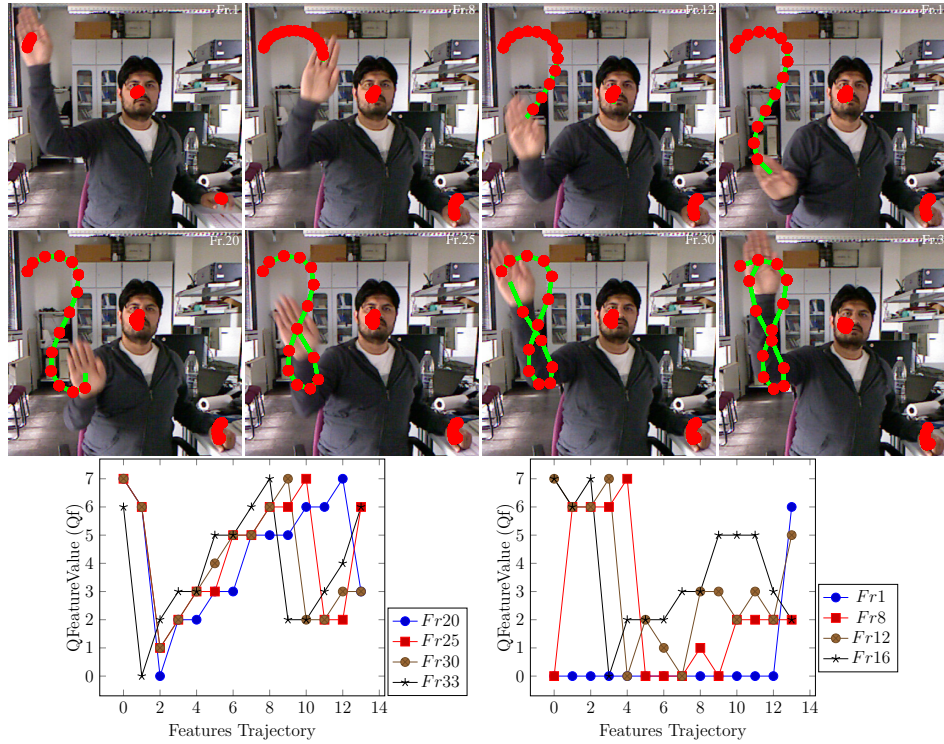


Figure 5.4: Gestural action when the subject is drawing *Gesture '8'* using Bezier descriptor ($N = 15$). In the sequence, Bezier points (marked as red points) are presented whereas the graphs present extracted quantized features with trajectories. Left graph shows the features of detected *Gesture '8'* whereas in right graph, no gesture is detected *Gesture '-1'*.

20, Fr 25, Fr 30, Fr 33 whereas the right graph shows Bezier descriptors with no detected gestural action (i.e., Fr 1 , Fr 8 , Fr 12 , Fr 16). In the proposed approach, HMM is modeled with Left-Right Banded Model (LRBM) topology using 14 states.

Fig. 5.5 presents the sequence with the subject drawing *Gesture '8'* with the graphs are showing trajectories of features. Like Fig. 5.4, hand centroid points features are transformed to Bezier points with $N = 5$ (i.e., marked as red color) which are then utilized to extract Bezier features ϑ and are binned to create Bezier descriptor. The above graph shows the quantized Bezier descriptor features generated from Bezier points with the ground truth data results in gestural action *Gesture '8'* at Fr 20, Fr 25, Fr 30, Fr 33 whereas the below graph shows the Bezier descriptor features where no detected gestural action (i.e., Fr 1, Fr 8, Fr 12, Fr 16). Similarly, we have experimented using

$N = 10$ Bezier points as shown in Fig. 5.6 and used the HMM classifier with 9 states. With the same motivation, experimental results are carried out for

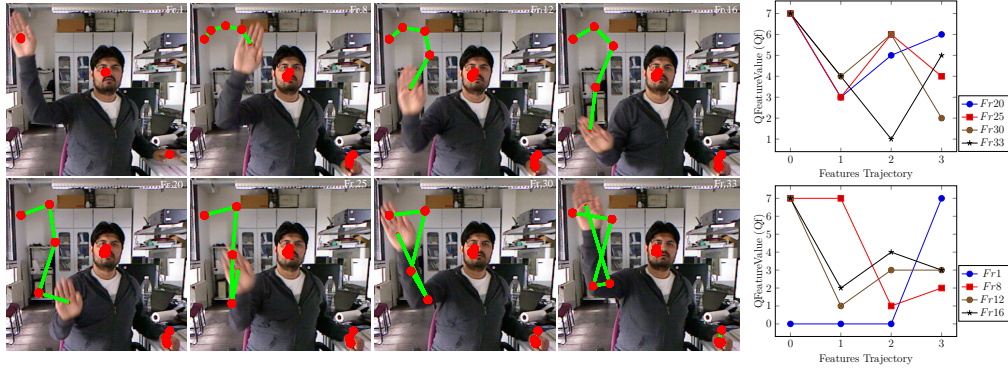


Figure 5.5: Gestural action when the subject is drawing *Gesture* ‘8’ with Bezier descriptor ($N = 5$). In the sequence, Bezier points (marked as red points) are presented whereas the graphs present the extracted quantized features with trajectories. The above graph shows the features of detected *Gesture* ‘8’ whereas in lower graph, no gesture is detected *Gesture* ‘-1’.

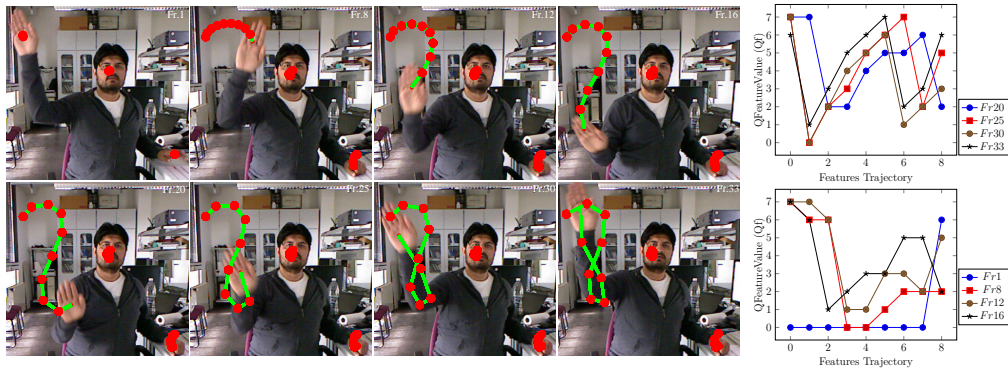


Figure 5.6: Gestural action when the subject is drawing *Gesture* ‘8’ using Bezier descriptor ($N = 10$). In the sequence, Bezier points (marked as red points) are presented whereas the graphs present extracted quantized features with trajectories. The above graph shows the features of detected *Gesture* ‘8’ whereas in lower graph, no gesture is detected *Gesture* ‘-1’.

higher number of Bezier curves points $N = 20$ and $N = 30$ as presented in Fig. 5.7 and Fig. 5.8. In these experiments, Bezier descriptors are modeled with higher number of Bezier points and are classified using HMM by employing higher number of states S . The results of employing higher number of states in HMM classification results in low recognition rate for the gestural actions.



Figure 5.7: Gestural action when the subject is drawing *Gesture* ‘8’ using Bezier descriptor ($N = 20$). In the sequence, Bezier points (marked as red points) are presented whereas the graphs present extracted quantized features with trajectories. Upper graph shows the features of detected *Gesture* ‘8’ whereas in lower graph, no gesture is detected *Gesture* ‘-1’.

Fig. 5.9 presents the classification results for different Bezier descriptors for the stated sequence along with ground truth data. It can be seen that Bezier descriptor $N = 15$ results in highest recognition rate for this sequence.

This comparison of different HMM models is carried out using LRBM and tested for different number of HMM states with the same input control points (i.e., centroid). These control points are modeled using Bezier curve from which the different features are extracted and binned down to construct the Bezier descriptors. In the following, performance of the Bezier descriptors is computed (i.e., $N = 5$, $N = 10$, $N = 15$, $N = 20$ and $N = 30$) against each other and with the centroid points (i.e., control points (CP)). For the control points, HMM is chosen with 30 states always. Moreover, we argue that the utilization of Bezier descriptors by fitting N points makes the proposed approach independent of HMM states model to be trained (i.e., extracted features length has to be same as the number of states in HMM which is difficult to maintain under varying frame rates). The proposed approach on IIKT-GP dataset achieves the overall accuracy of 98.3% (i.e., See Fig. 5.11) for gestural actions using Bezier descriptor $N = 15$. Fig. 5.11 provides a comparative analysis by adjusting N parameter (i.e., N is the number of points) for fitting Bezier curves.

Fig. 5.10 presents the confusion matrix of gestural symbols with Bezier descriptors $N = 5$ (i.e., upper graph - lowest classification rate (36.7%)) and $N = 15$ (i.e., lower graph - highest classification rate (98.3%)). It can be

seen that the first graph contains higher mis-classification whereas the mis-classifications are very less among the gestural symbols in the second graph. The higher rate of classification indicates that Bezier descriptor is capable of recognizing the gestural symbols.

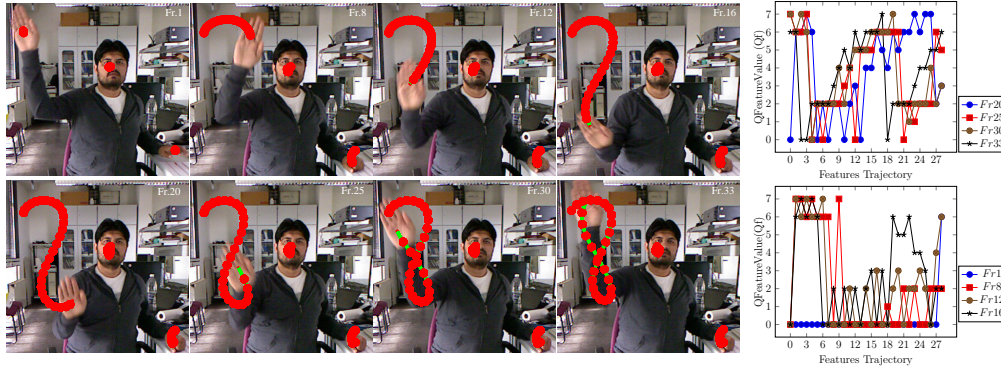


Figure 5.8: Gestural action when the subject is drawing *Gesture* ‘8’ using Bezier descriptor ($N = 30$). In the sequence, Bezier points (marked as red points) are presented whereas the graphs present extracted quantized features with trajectories. Upper graph shows detected *Gesture* ‘8’ features whereas in lower graph, no gesture is detected *Gesture* ‘-1’.

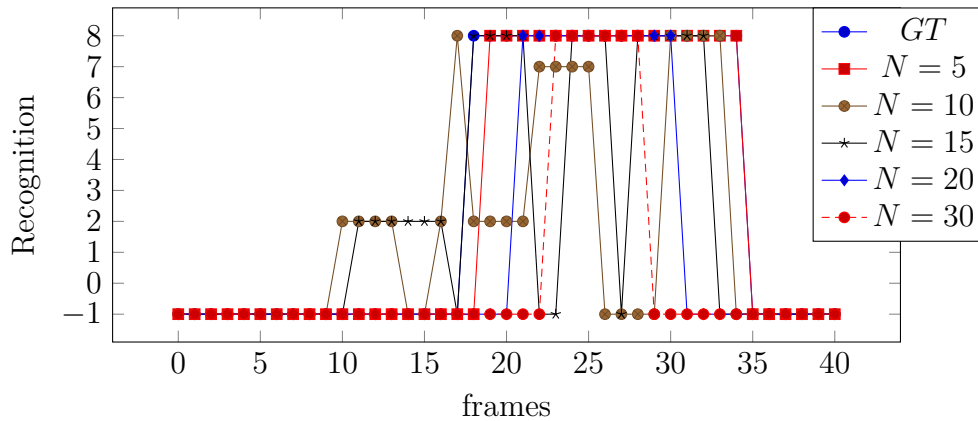


Figure 5.9: Classification of Bezier descriptors (i.e., $N = \{5, 10, 15, 20, 30\}$) using HMM for the sequence with recognized *Gesture* ‘8’ whereas *GT* defines the ground truth data. In the sequence, Bezier descriptor with $N = 15$ performs better than other descriptors. Y-axis in graph shows the classified gestural symbol whereas X-axis shows the sequence frames.

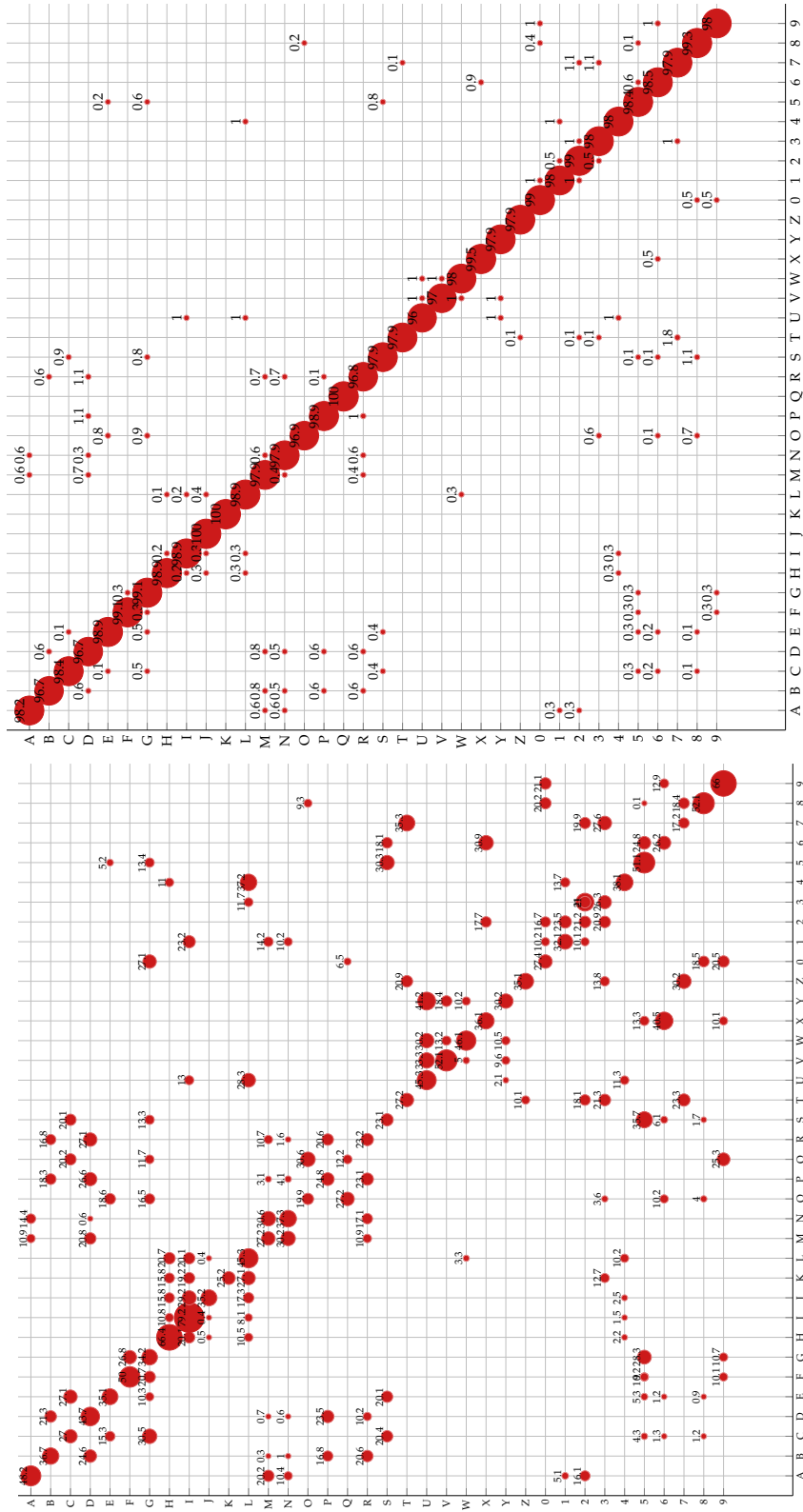


Figure 5.10: Confusion matrices of gesture symbols. Left graph shows the confusion matrix of Bezier descriptor $N = 5$ whereas the right graph presents the confusion matrix of $N = 15$. It can be seen that higher number of mis-classifications is occurred for Bezier descriptor $N = 5$ whereas lower number of mis-classifications is occurred for Bezier descriptor $N = 15$.

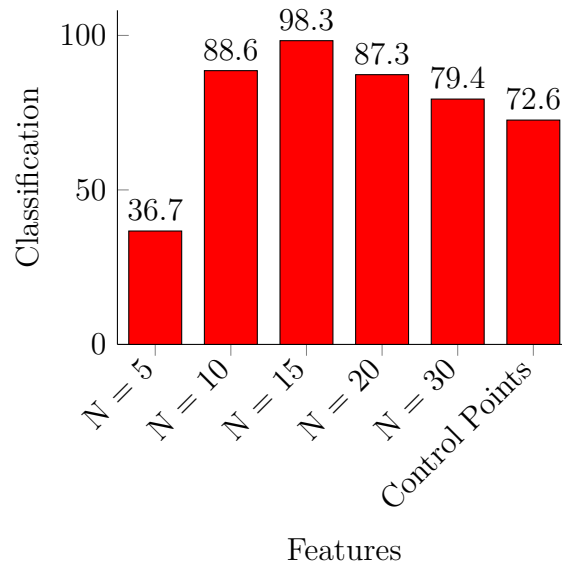


Figure 5.11: Classification rate of Bezier descriptors ($N = \{5, 10, 15, 20, 30\}$) and control points using HMM. The performance of Bezier descriptor $N = 15$ is the best amongst all.

5.2 Support Vector Machines

Support Vector Machines is a supervised learning approach for the optimal modelling of the data [129]. It learns the decision function and separates the data class to the maximum width. Basically, SVM works on two-class i.e., binary classification and is also extendable for multi-class problem. In the literature, there are two types of this extension. All-together approach deals with the optimization problem but it lacks scalability and faces optimization complexity. The second approach deals in binary fashion with multiple hyperplanes along with the combination into a single classifier. There are further two alternatives for this combination. The first one is based on one-against-all whereas other works as one-against-one.

Binary classification of SVM learns on the following principle:

$$c(x) \in \{-1, 1\} \quad (5.24)$$

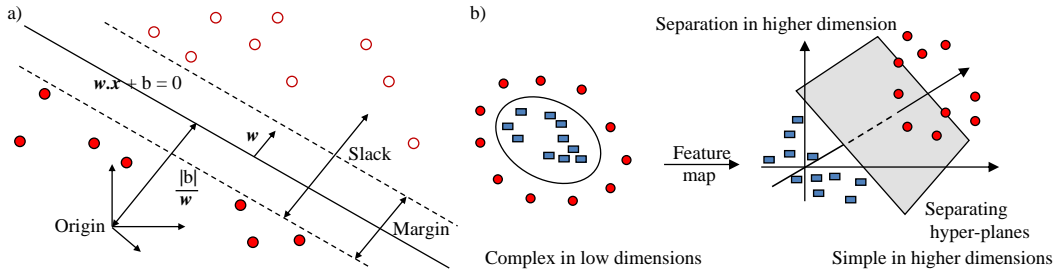


Figure 5.12: a) Margin of the hyper-plane. b) Mapping from input data to a richer feature space through kernel function.

The SVM's linearly learned decision function $f(x)$ is described as:

$$f(x) = \text{sign}(w \cdot x + b) \quad (5.25)$$

Where w is a weight vector while b is the threshold and x is the input sample. SVM learner defines the hyper-planes for the data where maximum margin is found between these hyper-planes. Due to the maximum separation of hyper-planes, it is also considered as a margin classifier. Margin of the hyper-plane is the minimum distance between hyper-plane and the support vectors and this margin is maximized. It can be formulated as:

$$\gamma = \frac{2}{\|w\|} \quad (5.26)$$

where γ is margin of the hyper-plane. Maximization of the margin of the hyper-plane is depicted in Fig. 5.12 a). Moreover, SVM maps input data into high dimension domain where it is utmost linearly separable as shown in Fig. 5.12 b). This mapping does not affect the training time because of implicit dot product and kernel trick [129, 130]. This is also a reason that SVM is a well suited classifier when features are large in number because they are robust to the curse of dimensionality. Kernel function [131] is the computation of the inner product $\phi(x) \cdot \phi(y)$ directly from the input. One of the characteristics of using the kernel is that there is no need to explicitly represent the mapped feature space. Kernel function is mathematically described as follows:

$$K(x, y) = \phi(x) \cdot \phi(y) \quad (5.27)$$

Following are some of the kernel functions which are commonly used to convert

the input features into new feature space [131].

- Linear kernel

$$K(x, y) = (x \cdot y) \quad (5.28)$$

- RBF Gaussian kernel

$$K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}} \quad (5.29)$$

- Polynomial kernel (homogeneous)

$$K(x, y) = (x \cdot y)^d \quad (5.30)$$

- Polynomial kernel (inhomogeneous)

$$K(x, y) = (x \cdot y + 1)^d \quad (5.31)$$

- Sigmoid kernel

$$K(x, y) = \tanh(\kappa x \cdot y + c) \quad (5.32)$$

where K is a scaling factor while c is a shifting factor that controls the mapping. As discussed above, *SVM* outputs only the class labels for the input sample as output but not the probability information for the classes. Lin et al. [132] describes a method to compute the class probabilities using *SVM*. Chang et al [133] developed a library (LIBSVM) which provides the tools for the *SVM* functionalities including class probability estimation. In the proposed approach, features in Eq. 4.14 are used to train and test the *SVM* classifier for detecting hand postures whereas Table 4.3 provides the feature combinations for posture analysis.

5.2.1 Categorization Results and Analysis

The posture symbols are categorized into different groups depending upon the detected fingers (see Section 4.4.3 and Table 4.2). As some ASL posture alphabets and numbers have very similar shape and structure (e.g., Posture ‘D’ and ‘1’, Posture ‘V’ and ‘2’, Posture ‘W’ and ‘3’ with minor thumb position differences), therefore, in the proposed approach, *ASL* alphabets and numbers are put in two separate classes for *SVM* classifier. In the following,

description about ASL alphabets and numbers are presented.

ASL Alphabets: In Table 4.2, 20 alphabets are categorized into four groups according to the fingertips and are presented as follows:

- *Group 1:* In this group, ASL alphabets consists of posture symbols A, B, E, F, O, X . The mis-classifications in this group is between ‘A’ and ‘E’ due to the correlation of the features in the second moments and geometrical properties. However, the third moment features are different and it results in recognition of these symbols. Moreover, the combination of features results in the reduction of mis-classifications. The second case are the posture symbols ‘B’ and ‘F’ which have correlated features because of the similar shapes. In this case, the second moment features are highly correlated but the third moments and geometrical features are different by which SVM discriminates these symbols correctly. Table 5.2 presents the confusion matrix and feature comparison of statistical and geometrical features along with the fusion of different features to enhance the performance of the posture recognition system. It can be seen that the feature fusion enhances the performance of the proposed approach as shown in the Table 5.2.
- *Group 2:* ASL alphabets in this group are $A, B, D, F, G, H, I, R, U$. Table 5.3 shows the confusion matrix of posture symbols with one detected fingertip. The mis-classifications in this group occur between the posture symbols ‘A’ and ‘F’ and between symbols ‘B’ and ‘H’ due to the correlated features in the third moments. Moreover, the geometrical features are highly correlated among the posture symbols ‘B’, ‘D’, ‘H’, ‘I’ and ‘U’. Moreover, ‘R’ and ‘U’ have high coherence in the second moment features. These results are presented in the confusion matrices in Table. 5.3 where the mis-classifications can be seen. Moreover, the fusion of posture features and its classification results using SVM are presented with confusion matrix to evaluate the performance and to represent the outcome of feature fusion.
- *Group 3:* ASL alphabets in this group are C, K, L, P, Q, V, Y and the confusion matrix in Table 5.4 shows the performance of the statistical and geometrical features in which the two fingertips are detected.

Moreover, during the experimentation process in this group, the highest misclassification occurs between symbols ‘P’ and ‘Q’ because of the shape and geometry of these posture signs. Besides, the statistical features in this group lies in the similar range and thus possesses a strong correlation which leads to the misclassification between them.

- *Group 4*: In this group, ASL alphabet ‘W’ is the only symbol and is always classified when three fingertips are detected.

ASL Numbers: In the proposed approach, ASL numbers (0 to 9) are categorized into six groups which contains from no finger to all five detected fingers as shown in Table 4.2. Moreover, Table 5.5 presents the ASL Numbers with their classification outcomes for the statistical and geometrical properties.

- *Groups 1,2,3,5,6*: These groups contain only one symbol, so the classification using SVM is dependent on the detected fingers.
- *Group 4*: This group contains the posture numbers with 3 detected fingers namely 3, 6, 7, 8, 9. In this group, the highest mis-classification rate occurs between the posture symbols 7, 8, 9 because of their similar shape and structure.

5.2.2 Experimental Results

The experimental setup for the posture recognition involves the tasks of data acquisition, hand detection, posture feature extraction and classification. Moreover, the applicability of our proposed system is demonstrated on real situations where the postures are recognized satisfying the criteria of ease, flexibility and naturalness. The proposed framework runs with real-time processing at 25 *fps* on Intel Processor 2.83GHz, 4 cores hardware configuration with 480×640 pixels image resolution. The experiments are conducted on 15 video observations per posture (i.e., approximately 50000 samples) of 6 subjects performing various hand postures wearing short-to-long sleeves and the posture dataset contains the finger spelling 30 posture symbols namely *A, B, C, D, E, F, G, H, I, K, L, O, P, Q, R, U, V, W, X, Y* and $0 \rightarrow 9$. However, as the posture symbols are static hand postures so, the dynamic postures *J, Z* are not included in the posture dataset. It is worth to mention that the

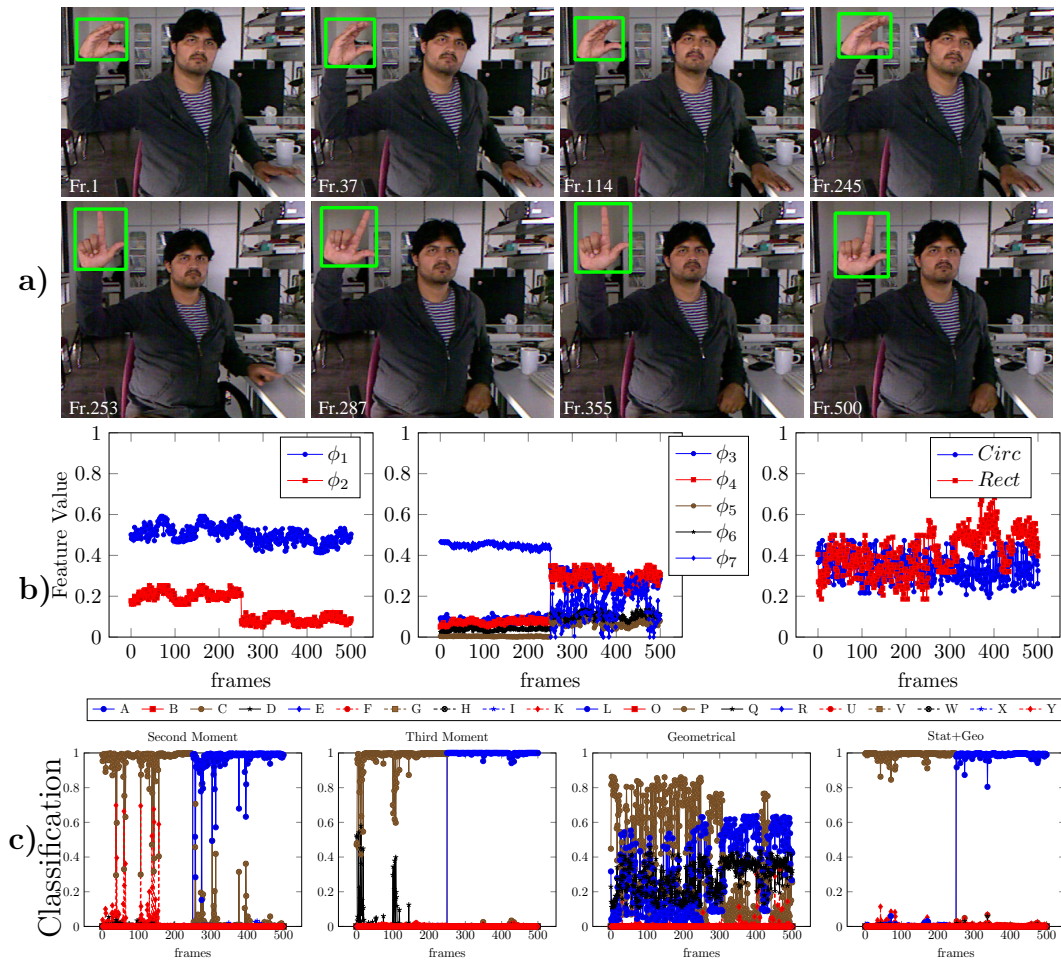


Figure 5.13: a) Sample sequence images shows the hand posture signs 'C' and 'L' in bounding boxes detected through skin segmented process at various time instances ('C' at *Fr. 1*, *Fr. 37*, *Fr. 114*, *Fr. 245* and 'L' at *Fr. 253*, *Fr. 287*, *Fr. 355*, *Fr. 500*). b) Graphs represent the normalized feature vector set (i.e., left - statistical features second-order moment, middle - statistical features third-order moment and right - geometrical features) c) Graphs represent the classified outcome for features (i.e., 1- statistical features second-order moment, 2- statistical features third-order moment, 3- geometrical features, 4- statistical and geometrical features).

training data are entirely different from the test data. In the proposed approach, posture symbols are extensively tested for translation, rotation and scaling properties. Additionally, the experimental results for the classification gives us an insight about the effect of categorization in finger spelling posture symbols.

In the sequence 5.13, posture sign *Posture* ‘C’ is detected from frames 1 to 249 and then *Posture* ‘L’ is detected from the frames 250 to 500. Fig. 5.13 a) The original frames with detected blobs from skin color segmentation are presented whereas the statistical ($\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \psi_7$) and geometrical features (*Circ, Rect*) are presented in Fig. 5.13 b). These posture signs are classified after the fingertip categorization process which calculates the curvatures to detect the fingers as described in Section. 4.3. In Fig. 5.13 c), the analysis of statistical and geometrical features is presented in the graph for the posture symbols. Fig. 5.13 c)-1 presented the results of second moment (i.e., ψ_1, ψ_2) whereas the third moments (i.e., $\psi_3, \psi_4, \psi_5, \psi_6, \psi_7$) results are shown in Fig. 5.13 c)-2. The geometrical features (*Circ, Rect*) are presented in Fig. 5.13 c)-3 whereas the concatenation of statistical and geometrical features are presented in Fig. 5.13 c)-4. In these graphs, X-axis shows the frames whereas Y-axis shows the classification probabilities of all posture symbols in this sequence. It can be seen from the results that the combination of statistical and geometrical features increase the classification rate and has the dominated performance over the other features recognition results (i.e., second moment, third moment and geometrical features) whereas geometrical features for the alphabets ‘C’ and ‘L’ are highly coherent with other symbols which results in mis-classifications. The second order moments for alphabets ‘C’ and ‘L’ have higher mis-classifications with other signs in Group 3 (i.e., with alphabets ‘K’, ‘V’). The thid order moments in this sequence has only few mis-classifications with symbol ‘Q’.

We have tested the proposed approach on IIKT-GP database with the overall accuracy of 97.8% as shown in Fig. 5.14 for posture symbols. In this graph, a comparison is performed utilizing different moments (i.e., Zernike moments (see Appendix A.1) and Hu-moments) and geometrical features along with the reduced features set resulted from Principal Component Analysis (PCA) [134]. PCA is a statistical technique to find the principal components of data. In various applications such as face recognition, hand gesture and posture recognition and image compression, PCA is used to reduce the features set. The

Table 5.1: Average processing-time in milliseconds (640×480) for the proposed approach of Gesture (G) and Posture (P) recognition

Modules	Modality	Processing Time ms
Segmentation	G&P	18.8
Hand Detection	G&P	2.3
Feature Extraction	G	4.7
	P	6.2
Classification	G	2.6
	P	2.7
Total time in msec	G&P	37.3

functioning of PCA is based on the concept of deconstructing the set of data points into eigenvectors and eigenvalues, both exists in pairs [135]. In PCA, for every eigenvector, there is a corresponding eigenvalue where the vectors denote the direction and values contain the variance in underlying direction. As a result, the highest value of eigenvector and eigenvalue is the principal component. In the comparison, PCA is applied to reduce the feature set of statistical and geometrical features from 9 features to 5 and 6 K-components as it represents 89% and 95% of the data respectively. However, the feature set computed from PCA do not have a significant impact on the classification results from SVM with 5 and 6 K-components. It is due to the fact that in PCA space, the features are not discriminative enough and therefore, SVM results in mis-classifications of symbols.

In Tables 5.2, 5.3, 5.4 and 5.5, we have provided a comparative analysis of different moments and geometrical features for posture recognition and observe that the performance of statistical with geometrical features is superior amongst all as shown in Fig. 5.14.

5.3 Summary and Conclusion

In this chapter, feature analysis and classification is presented for gesture and posture recognition systems along with experimental results. Firstly, HMM is presented which is followed by the experimental results and performance analysis for gesture recognition. Secondly, SVM are explained for the recognition of posture symbols along with the experimental results and analysis. Moreover, the categorization of the posture symbols and its effect on the posture

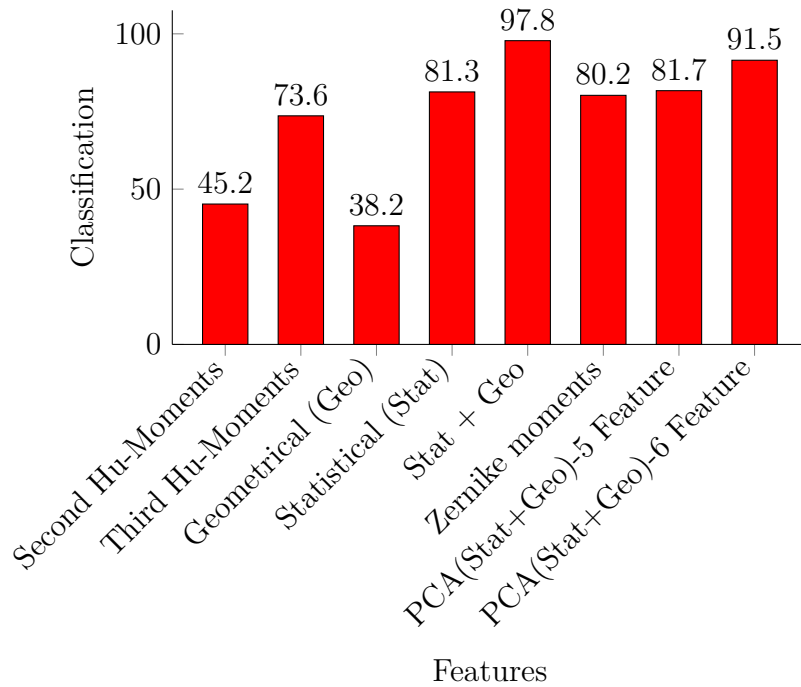


Figure 5.14: Classification rate of the posture symbols with and without fingertip detection process. It can be seen that by fusing statistical and geometrical features, the posture recognition is significantly improved. Moreover, Zernike moments and PCA feature set (i.e., reduction from statistical and geometrical features (9 features) to 5 and 6 K-components) are also presented.

recognition is presented. The experiments conducted on IKT-GP dataset considering different features for the performance of proposed gesture and posture recognition approaches.

Table 5.2: Confusion Matrix of Statistical and Geometrical Features - Group 1 with no Fingertip Detected

Sign	Features	FT	A	B	E	F	O	X
A	$\{\psi_1, \psi_2\}$	✓	30.2	7.1	28.7	3.8	3.8	26.4
	$\{\psi_3, \dots, \psi_7\}$	✓	68.8	9.1	0.0	20.1	0.7	1.3
	$\{Circ, Rect\}$	✓	35.2	2.3	20.8	3.4	7.1	31.2
	$\{\psi_1, \dots, \psi_7\}$	✓	78.2	0.0	2.6	14.2	1.6	3.4
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	98.2	0.0	0.2	1.3	0.3	0.0
B	$\{\psi_1, \psi_2\}$	✓	2	47.8	11.4	30.4	3.8	4.6
	$\{\psi_3, \dots, \psi_7\}$	✓	4	71.1	22.9	0.0	0.8	1.2
	$\{Circ, Rect\}$	✓	0.0	26.2	31.7	28.4	3	10.7
	$\{\psi_1, \dots, \psi_7\}$	✓	0.7	81.1	8.9	9.1	0.2	0.0
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.1	96.8	1.5	1.5	0.1	0.0
E	$\{\psi_1, \psi_2\}$	✓	26.3	10.1	30.3	11.3	0.8	22.2
	$\{\psi_3, \dots, \psi_7\}$	✓	0.0	20.1	78.6	1.3	0.0	0.0
	$\{Circ, Rect\}$	✓	13.1	21.0	37.8	28.1	0.0	0.0
	$\{\psi_1, \dots, \psi_7\}$	✓	0.0	11.1	85.2	0.0	0.0	3.7
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	0.4	98.9	0.0	0.0	0.7
F	$\{\psi_1, \psi_2\}$	✓	1	27.0	12	56.3	2.0	1.7
	$\{\psi_3, \dots, \psi_7\}$	✓	20.1	3.7	6.5	61.2	0.0	8.5
	$\{Circ, Rect\}$	✓	0.4	27.4	34.1	31.3	6.8	0.0
	$\{\psi_1, \dots, \psi_7\}$	✓	7.7	10.7	3.3	78.3	0.0	0.0
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.2	1.8	0.3	97.7	0.0	0.0
O	$\{\psi_1, \psi_2\}$	✓	0.0	6.5	0	0	68.3	25.2
	$\{\psi_3, \dots, \psi_7\}$	✓	0.0	4.9	5.1	0	80	10
	$\{Circ, Rect\}$	✓	0.4	13.2	10.7	15.2	57.3	3.2
	$\{\psi_1, \dots, \psi_7\}$	✓	0.0	2.4	0.0	0.0	87.2	10.4
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	0.2	0.0	0.0	99.1	0.7
X	$\{\psi_1, \psi_2\}$	✓	19.1	10	0.0	0.0	29.6	41.3
	$\{\psi_3, \dots, \psi_7\}$	✓	0.4	5.7	0.0	1.7	12.1	80.1
	$\{Circ, Rect\}$	✓	44.4	5.3	4.7	1.8	3.2	40.6
	$\{\psi_1, \dots, \psi_7\}$	✓	2.5	0.0	0.0	0.0	9.2	88.3
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	1.3	0.0	0.0	0.0	1.6	97.1

Table 5.3: Confusion Matrix of Statistical and Geometrical Features - Group 2 with One Fingertip Detected

Sign	Features	FT	A	B	D	F	G	H	I	R	U
A	$\{\psi_1, \psi_2\}$	✓	40.8	6.3	3.4	2.5	4.9	5.6	30.5	3.1	2.9
	$\{\psi_3, \dots, \psi_7\}$	✓	75.2	12.2	1.2	1.0	0.6	0.2	9.2	0.2	0.2
	$\{Circ, Rect\}$	✓	65.5	0.0	15.6	0.0	0.0	5.5	13.4	0.0	0.0
	$\{\psi_1, \dots, \psi_7\}$	✓	80.7	7.3	0.2	0.4	0.4	3.2	7.5	0.1	0.2
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	96.1	1.3	0.0	0.0	1.2	0.2	1.2	0.0	0.0
B	$\{\psi_1, \psi_2\}$	✓	7.2	57.3	11.2	16.3	0.2	7.8	0.0	0.0	0.0
	$\{\psi_3, \dots, \psi_7\}$	✓	0.0	72.3	0.0	10.7	7.7	4.6	4.7	0.0	0.0
	$\{Circ, Rect\}$	✓	0.0	46.3	0.0	25.8	14.5	23.4	0.0	0.0	0.0
	$\{\psi_1, \dots, \psi_7\}$	✓	0.0	78.8	0.0	9.2	6.5	4.2	1.3	0.0	0.0
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	97.2	0.0	2.2	0.2	0.2	0.2	0.0	0.0
D	$\{\psi_1, \psi_2\}$	✓	0.0	11.3	70.5	7.3	2.3	8.6	0.0	0.0	0.0
	$\{\psi_3, \dots, \psi_7\}$	✓	0.0	0.0	76.8	0.0	0.0	0.0	0.0	10.1	13.1
	$\{Circ, Rect\}$	✓	17.4	0.0	51.2	0.0	0.0	10.2	21.2	0.0	0.0
	$\{\psi_1, \dots, \psi_7\}$	✓	0.0	0.0	84.7	0.0	0.0	0.0	0.0	7.5	7.8
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	0.0	96.8	0.0	0.0	0.0	0.0	2.2	1.0
F	$\{\psi_1, \psi_2\}$	✓	0.0	21.3	0.0	56.2	0.1	21.9	0.0	0.1	0.4
	$\{\psi_3, \dots, \psi_7\}$	✓	1.3	13.2	0.0	68.5	0.0	17.0	0.0	0.0	0.0
	$\{Circ, Rect\}$	✓	0.0	19.9	0.0	52.8	14.5	12.8	0.0	0.0	0.0
	$\{\psi_1, \dots, \psi_7\}$	✓	0.2	10.1	0.0	78.3	0.0	11.4	0.0	0.0	0.0
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.1	1.1	0.0	97.5	0.0	1.3	0.0	0.0	0.0
G	$\{\psi_1, \psi_2\}$	✓	0.0	10.4	0.0	10.3	55.8	6.2	15.1	2.2	0.0
	$\{\psi_3, \dots, \psi_7\}$	✓	0.0	0.0	1.2	0.0	80.5	6.5	11.8	0.0	0.0
	$\{Circ, Rect\}$	✓	0.0	8.8	0.0	26.8	54.3	10.1	0.0	0.0	0.0
	$\{\psi_1, \dots, \psi_7\}$	✓	0.0	0.0	0.0	0.0	90.8	2.2	7.0	0.0	0.0
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	0.0	0.0	0.0	99.4	0.1	0.5	0.0	0.0
H	$\{\psi_1, \psi_2\}$	✓	0.1	19.2	0.0	23.6	0.2	54.4	0.2	1.2	1.1
	$\{\psi_3, \dots, \psi_7\}$	✓	0.0	10.0	0.0	0.0	0.0	65.5	5.1	19.4	0.0
	$\{Circ, Rect\}$	✓	0.0	26.4	0.0	13.8	14.7	45.1	0.0	0.0	0.0
	$\{\psi_1, \dots, \psi_7\}$	✓	0.0	5.0	0.0	6.3	0.0	73.8	2.8	12.1	0.0
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	0.7	0.0	1.0	0.0	91.8	1.3	5.2	0.0
I	$\{\psi_1, \psi_2\}$	✓	0.0	0.0	0.1	6.2	24.2	3.8	65.7	0.0	0.0
	$\{\psi_3, \dots, \psi_7\}$	✓	0.0	0.0	0.0	12.3	0.0	0.0	80.2	1.8	5.7
	$\{Circ, Rect\}$	✓	4.0	0.0	34.2	0.0	0.0	0.0	61.8	0.0	0.0
	$\{\psi_1, \dots, \psi_7\}$	✓	0.0	0.0	0.0	6.7	0.0	0.0	90.8	0.3	2.2
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	0.0	0.2	0.0	0.0	0.0	99.6	0.1	0.1
R	$\{\psi_1, \psi_2\}$	✓	5.6	0.0	3.0	4.5	3.9	1.7	0.0	70.8	10.5
	$\{\psi_3, \dots, \psi_7\}$	✓	0.0	0.0	10.8	0.0	0.0	0.0	0.0	82.4	6.8
	$\{Circ, Rect\}$	✓	0.0	0.0	0.0	0.0	0.0	11.3	1.9	62.1	24.7
	$\{\psi_1, \dots, \psi_7\}$	✓	0.0	0.0	2.1	0.0	0.0	0.9	0.0	92.5	4.5
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	0.0	0.3	0.0	0.0	0.2	0.0	98.3	1.2
U	$\{\psi_1, \psi_2\}$	✓	4.4	1.1	3.0	1.2	0.2	1.3	6.5	17.1	65.2
	$\{\psi_3, \dots, \psi_7\}$	✓	0.0	0.0	15.2	0.2	0.2	0.0	0.2	13.1	71.1
	$\{Circ, Rect\}$	✓	0.0	0.0	0.0	0.0	0.0	12.5	0.1	32.1	55.3
	$\{\psi_1, \dots, \psi_7\}$	✓	0.0	0.0	4.2	0.0	0.2	0.0	0.0	13.5	82.1
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	0.0	1.2	0.0	0.1	0.0	0.0	2.1	96.6

Table 5.4: Confusion Matrix of Statistical and Geometrical Features - Group 3 with Two Fingertips Detected

Sign	Features	FT	C	K	L	P	Q	V	Y
C	$\{\psi_1, \psi_2\}$	✓	51.7	12.3	17.2	4.5	4.3	5.2	4.8
	$\{\psi_3, \dots, \psi_7\}$	✓	75.3	8.3	7.9	2.2	2.1	4.2	0
	$\{Circ, Rect\}$	✓	52.1	3.7	18.5	4.3	14.3	4.1	3.4
	$\{\psi_1, \dots, \psi_7\}$	✓	87.3	4.3	2.1	3.2	0	2.4	0.7
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	98.7	0	0.3	0.7	0	0.3	0
K	$\{\psi_1, \psi_2\}$	✓	8.7	42.2	5.3	2.8	0	38.2	2.8
	$\{\psi_3, \dots, \psi_7\}$	✓	10.1	78.3	3.1	2.3	0	5.4	0.8
	$\{Circ, Rect\}$	✓	4.5	59.3	6	0	0	30.2	0
	$\{\psi_1, \dots, \psi_7\}$	✓	2.7	87.5	1.2	2	0	6.1	0.5
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.2	98.3	0.3	0	0	1.2	0
L	$\{\psi_1, \psi_2\}$	✓	23.5	10.2	53.1	7.3	1.2	1.9	2.8
	$\{\psi_3, \dots, \psi_7\}$	✓	5.8	7.1	76.7	0	2.2	3.3	4.9
	$\{Circ, Rect\}$	✓	28.3	7.3	43.3	0	0	10.7	10.4
	$\{\psi_1, \dots, \psi_7\}$	✓	2.3	3.3	90.4	0	1	0.8	2.2
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.4	0	98.5	0	0.7	0	0.4
P	$\{\psi_1, \psi_2\}$	✓	2.5	11	0	49.3	37.2	0	0
	$\{\psi_3, \dots, \psi_7\}$	✓	3.1	3.4	0	77.3	16.2	0	0
	$\{Circ, Rect\}$	✓	4.3	9.4	0	50.3	36	0	0
	$\{\psi_1, \dots, \psi_7\}$	✓	0	2.4	0	87.4	10.2	0	0
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0	0	0	98.7	1.3	0	0
Q	$\{\psi_1, \psi_2\}$	✓	2	2	0	40.8	52.9	2.3	0
	$\{\psi_3, \dots, \psi_7\}$	✓	4.1	7.5	0	18.1	67.3	1.9	1.1
	$\{Circ, Rect\}$	✓	15.6	8.1	0	31	45.3	0	0
	$\{\psi_1, \dots, \psi_7\}$	✓	0	4.7	0	13.7	80.3	1.3	0
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0	0	0	3.8	96.2	0	0
V	$\{\psi_1, \psi_2\}$	✓	3.1	40.9	5.4	0	0	45.2	5.4
	$\{\psi_3, \dots, \psi_7\}$	✓	0.1	21	4.3	0	0	67.8	7.8
	$\{Circ, Rect\}$	✓	8.4	27.8	5.7	0	0	53.8	4.3
	$\{\psi_1, \dots, \psi_7\}$	✓	1.1	10.4	3	0	0	82.1	3.4
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.2	0.8	0	0	0	98.6	0.4
Y	$\{\psi_1, \psi_2\}$	✓	10.2	3.2	0	0	0	14.2	72.4
	$\{\psi_3, \dots, \psi_7\}$	✓	7.3	6.2	0	0	0.2	10.9	75.4
	$\{Circ, Rect\}$	✓	4.7	29.4	0	0	0.2	12.3	53.4
	$\{\psi_1, \dots, \psi_7\}$	✓	1.1	1.9	0	0	0	7.8	89.2
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0	0	0	0	0	0.7	99.3

Table 5.5: Confusion Matrix of Statistical and Geometrical Features - Group 4 with Three Fingertips Detected

Nr.	Features	FT	3	6	7	8	9
3	$\{\psi_1, \psi_2\}$	✓	46.1	18.5	16.8	10.2	8.4
	$\{\psi_3, \dots, \psi_7\}$	✓	72.6	9.1	8.2	6.6	3.5
	$\{Circ, Rect\}$	✓	46.4	21.3	18.1	5.1	9.1
	$\{\psi_1, \dots, \psi_7\}$	✓	86.3	4.5	2.4	3.2	3.6
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	98.2	0.6	0.4	0.7	0.1
6	$\{\psi_1, \psi_2\}$	✓	23.4	38.2	12.2	7.9	18.3
	$\{\psi_3, \dots, \psi_7\}$	✓	12	70.4	5.3	6.2	6.1
	$\{Circ, Rect\}$	✓	12.9	36.3	27.4	8.5	16.9
	$\{\psi_1, \dots, \psi_7\}$	✓	3.7	90.0	2.1	1.8	2.4
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.6	98.6	0.1	0.1	0.6
7	$\{\psi_1, \psi_2\}$	✓	12.6	25.3	33.4	15.3	13.4
	$\{\psi_3, \dots, \psi_7\}$	✓	13.4	7.3	69.8	7.1	2.4
	$\{Circ, Rect\}$	✓	8.3	7.9	40.2	20.3	23.3
	$\{\psi_1, \dots, \psi_7\}$	✓	2.6	3.2	88.2	2.4	3.6
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.2	0.4	99.0	0.3	0.1
8	$\{\psi_1, \psi_2\}$	✓	11.9	6.9	22.3	40.5	18.4
	$\{\psi_3, \dots, \psi_7\}$	✓	3.2	3.3	8.5	73.4	11.6
	$\{Circ, Rect\}$	✓	12.3	8.7	19.9	34.5	24.6
	$\{\psi_1, \dots, \psi_7\}$	✓	2.6	2.4	3.7	84.6	6.7
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.0	0.0	0.5	98.9	0.6
9	$\{\psi_1, \psi_2\}$	✓	7.3	10.3	21.4	23.7	37.3
	$\{\psi_3, \dots, \psi_7\}$	✓	5.1	7.2	8.3	5.1	74.3
	$\{Circ, Rect\}$	✓	9.4	4.3	27.4	19.3	39.6
	$\{\psi_1, \dots, \psi_7\}$	✓	3.4	6.4	5.1	3.9	81.2
	$\{\psi_1, \dots, \psi_7, Circ, Rect\}$	✓	0.2	0.7	0.4	0.6	98.1

Integration and Inferences

This chapter describes the suggested methodology of integration of gesture and posture modalities along with inference mechanism to extract meaningful expressions. Section 6.1 presents the underlying integration concept for the gesture and posture recognition. Based on this concept, a particle filter system is proposed in Section 6.2 and is followed by interpretation and inference mechanism in Section 6.3. After that, experimental results are presented in Section 6.4 along with the analysis and is followed by sketching the summary and conclusion in Section 6.5.

6.1 Concept

Integration of different modalities aims to increase the robustness of a system and improves its performance in an unconstrained environment. In this context, researchers opted various modalities to enhance the performance specially in the field of biometrics for the security, forensics and identification processes. In multi-modal biometric systems, fusion can take place at different levels which includes sample level, feature level, match score level and decision level fusion [136].

In the multi-modal biometric system, combined multiple cues (face and voice authentication) are used by [137] for person identification. Moreover, Chang et al. [138] proposed a face recognition system to fuse 2D and 3D face information to improve the recognition rate. Wu et al. [139] proposed a multi-model system to combine the face recognition system with gait recognition to detect the humans. Wang et al. [140] proposed a hybrid scheme composed of linear discriminant analysis and Radial basis function to fuse face and iris biometrics. Frischholtz and Dieckmann [141] developed BioID system to fuse face, voice, and lip movements. In the same context, Choudhury et al. [142] combine unconstrained audio and video using Bayesian networks. Exploiting the hand features, Ross and Jain [136] combine fingerprints, hand geometry

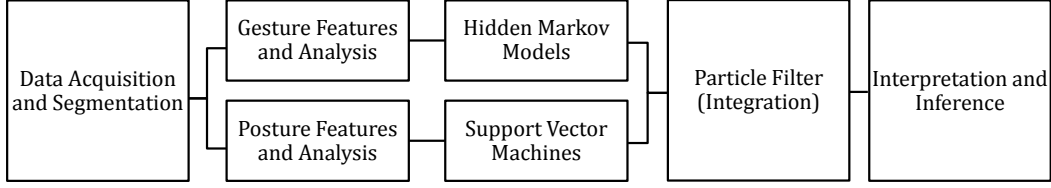


Figure 6.1: Process flow of the proposed framework for integration, interpretation and inference.

and face biometrics using the weighted sum rule. Similarly, Kumar et al. [143] performed fusion at feature level and match score level to combine the palm prints and hand geometrical features. To sum up, it is observed that the main motivation of exploiting different modalities is to achieve better performance and to cop with the limitations of uni-modal approach.

In the proposed approach (see Fig. 6.1), the basic idea is the interpretation of multiple signs driven from different modalities to infer meaningful expressions. To achieve this objective, the gesture and posture modalities are combined in a particle filter system at decision level to allow the inference of new symbols at any instance of time. The proposed integration I of gesture and posture recognition is formulated as:

$$\mathcal{I} = \langle \alpha_{gstr} \bullet \mathcal{R}_{hmm} \rangle \cap \langle \alpha_{pstr} \bullet \mathcal{R}_{svm} \rangle \quad (6.1)$$

where \mathcal{R}_{hmm} and \mathcal{R}_{svm} are the classification outcome of gesture and posture system. α_{gstr} and α_{pstr} are the contribution-weights associated with gesture and posture system as the integration criteria. In the following section, contribution-weights for gesture and posture modalities through particle filter system is presented which results in the integration and inferences of new symbols.

6.2 Particle Filter System

Condensation algorithm [144]) is a form of Bayesian estimation which provides a way to compute the a-posteriori probability (i.e. contribution-weight) and allows an effective integration process due to recursive state estimation process. A system of particle filters (i.e. comprising of two separate particle filters (i.e., for gesture and posture)) is implemented for which the idea,

algorithm and its processes are explained as follows.

The key idea of particle filter (PF) is to approximate the probability density function using a collection of random samples with associated weights from classification probabilities. In the particle filter system, the classification outcomes of gesture and posture recognition are maintained as state vectors (i.e., 1-dimensional) represented by $x_{gstr} = [R_{hmm}]$ and $x_{pstr} = [R_{svm}]$ respectively.

6.2.1 Initialization

In the initialization phase, particles are generated which begins by obtaining a set of initialization observations as shown in Fig. 6.2 a), b). The parameters for each particle are then generated by sampling from Gaussian distribution describing the classification outcomes. In the proposed approach, the measurements (i.e., for gesture and posture) at each time instance t are described as $z_m = \{z_{gstr}, z_{pstr}\}$ where z_{gstr} and z_{pstr} are the measurements of gesture and posture modalities respectively. A set of particles in vector $S(n)$ is represented as follows:

$$S(n) = \{s_k^{(gstr)}, s_k^{(pstr)}\} \quad (6.2)$$

A set of N random points (i.e., 100) called particles x_k^n with weights w_k^n denotes the initial distribution of particles at time k for both gesture and posture systems¹. These particles are denoted as:

$$s_k^{(gstr|pstr)} = \{x_k^n, w_k^n\}_{n=1}^N \quad (6.3)$$

We propose a generic framework for probability distribution in the particle filter as follows:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mathbf{x} - z_m)^2}{2\sigma^2}} \quad (6.4)$$

where σ is the standard deviation for the gesture or posture modality. The above distribution is sampled for each new particle from the cumulative probability of each \mathbf{x} .

¹The same notation is used for both the particle filters (i.e., gesture and posture), except when stated otherwise.

6.2.2 Selection

In the selection step, factored sampling is used to select the particles based on their weights and is achieved by selecting a random index from the cumulative weight of particles. Consequently, the particles having weights closer to the peaks in the probability distribution are sampled many times whereas the samples with low values in probability distributions are discarded. For the new observations, we only keep the samples with weights more than the average weights whereas the samples falling below the average weights are discarded and are replaced with new samples using the initialization distribution.

6.2.3 Prediction

The prediction step reflects the underlying temporal behavioral model where the particles are moved to generate the hypothetical state (x_k) at time k is based on the previous state (x_{k-1}). This step involves the sampling from the state transition and is formulated as follows:

$$p^{(n)}(x_k|z_{k-1}) = p^{(n)}(x_k|x_{k-1})p^{(n)}(x_{k-1}|z_{k-1}) \quad (6.5)$$

where $p(x_k|z_{k-1})$ is a-priori probability, $p(x_{k-1}|z_{k-1})$ is the previous a-posteriori probability and $p(x_k|x_{k-1})$ is the state transition model.

6.2.4 Updation

In the updation step, we compute the contribution-weights (a-posteriori probability) through the computed a-priori probability $p(x_k|z_{k-1})$ and the likelihood $p(z_k|x_k)$ by incorporating the new measurement z_k extracted from the classification outcomes of gesture and posture recognition systems (i.e., where the propagation process follows the same process 6.4). The contribution-weights $\alpha_{gstr|pstr}$ or a-posteriori probability $p(x_k|z_k)$ for gesture and posture system are computed as follows:

$$\alpha_{gstr|pstr} = p(x_k|z_k) = \frac{\sum_{n=1}^N p^{(n)}(z_k|x_k)p^{(n)}(x_k|z_{k-1})}{\sum_{n=1}^N p^{(n)}(z_k|x_k)} \quad (6.6)$$

Using N values of $p(z_k|x_k)$, we have obtained a probability distribution for the state space at time instance k . Each sample is assigned a weight according

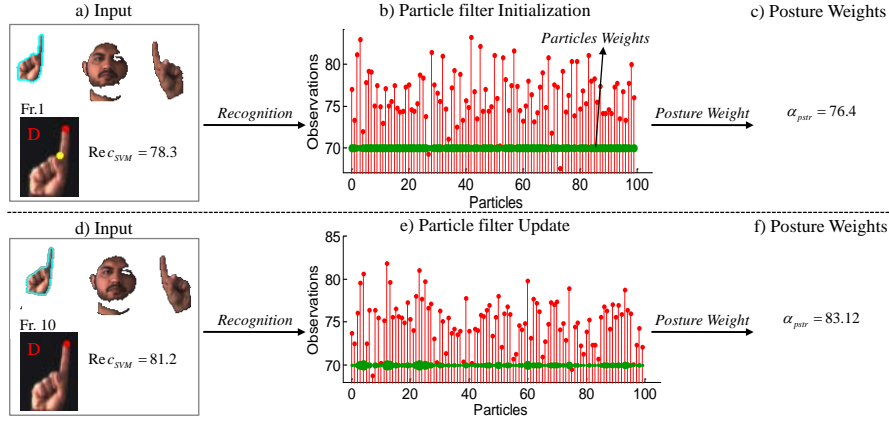


Figure 6.2: Particle filter process for the frames 1 and 10. a) Classification of ASL sign “D” with recognition outcome “78.3”. b) Particle filter weights initialization based on classification outcome. c) Posture contribution-weight. d) The classification of ASL sign “D” with recognition rate “81.2” for frame 10. e) Particle filter update based on classification outcome. f) Posture contribution-weight.

to its particular position in state space relative to observational density. In this way, we obtain the contribution-weights which defines the integration-criteria for the fusion of these systems. The same procedure is followed for each frame as presented in Fig. 6.2 d-f). The final integration is carried out when contribution-weights of gesture α_{gstr} and posture α_{pstr} signs satisfy the threshold ($T = 70\%$) at any time instance.

$$(\alpha_{gstr} | \alpha_{pstr}) \geq T \quad (6.7)$$

After obtaining the contribution-weights, we have used AND/OR combination for gesture and posture recognition signs. Integration I is formulated as:

$$I = \langle \alpha_{gstr} \bullet R_{hmm} \rangle \cap \langle \alpha_{pstr} \bullet R_{svm} \rangle \quad (6.8)$$

$$I = \langle \alpha_{gstr}^i \bullet R_{hmm}^i; i = 1 \dots m \rangle \cap \langle \alpha_{pstr}^j \bullet R_{svm}^j; j = 1 \dots n \rangle \quad (6.9)$$

In the suggested approach, integration of gesture and posture module interprets and infers when multiple posture symbols (described as n) are combined with a gesture symbol (m) or multiple gesture symbols are fused with a posture symbols (n). So, in the next section, we present the interpretation and

inference module for the fusion of gesture and posture modalities to generate meaningful expressions.

Table 6.1: Lexicon of Symbols

Symbols \Rightarrow Fruits	Symbols \Rightarrow Fruits
A \Rightarrow Apple, Apricot	N \Rightarrow Nectarine
B \Rightarrow Blueberry, Banana	O \Rightarrow Orange, Oval Kumquat
C \Rightarrow Cherry, Cantaloupe	P \Rightarrow Pear, Peach
D \Rightarrow Date, Dewberry	Q \Rightarrow Quince
E \Rightarrow Elderberry, Eggfruit	R \Rightarrow Raspberry, Rambutan
F \Rightarrow Fig, Farkleberry	S \Rightarrow Star Fruit, Strawberry
G \Rightarrow Grapes, Gooseberry	T \Rightarrow Tangerine, Tart Cherry
H \Rightarrow Honeymelon, Hackberry	U \Rightarrow Ugli Fruit, Uniq Fruit
I \Rightarrow Imbe	V \Rightarrow Voavanga
J \Rightarrow Jackfruit, Jambolan	W \Rightarrow Watermelon, Wolfberry
K \Rightarrow Kaffir Lime, Kiwi	X \Rightarrow Xigua
L \Rightarrow Lemon, Lychee	Y \Rightarrow Yunnan Hackberry
M \Rightarrow Mango, Melon	Z \Rightarrow Zinfandel Grapes

6.3 Interpretation and Inferences

After computing the contribution-weights for gesture and posture modalities, integration of these modalities for generating interpretations and inferences is the main objective. To achieve this goal, we consider the integration as a problem of regular language and mapped the recognition outcome over context free grammar (CFG) rules [145]. Before describing the concept of context specific interpretation and inference rules employed in this research, it is essential to first describe the proposed structure of language. The CFG grammar is defined in quadruple (i.e., 4-tuple) described as:

$$Grammar = \langle V, T, S, R \rangle \quad (6.10)$$

where V is the set of objects and contains non-terminals as well as terminals symbols, T is the set of terminals, S is start symbol and it is a subset of V (i.e., $S \in V$), and R is the set of production rules. The recognition outcomes are mapped on CFG rules(*) for the integration.

In CFG production rules, $\langle Pstr_A \rangle$ contains the set of posture alphabet signs, $\langle Gstr_A \rangle$ contains the set of gesture symbols, $\langle Pstr_N \rangle$ is the set of posture number set and $\langle Gstr_N \rangle$ contains the set of gesture number signs. Moreover,

Defs. Rules 1 Context Free Grammar (CFG)**Definitions and Rules :*

$$V = \{S, X, Y, AP, NP, AG, NG, Gstr_A, Gstr_N, Pstr_A, Pstr_N, \\ 0_p, 1_p, \dots, 9_p, a_g, b_g, \dots, z_g, a_p, b_p, \dots, z_p, 0_g, 1_g, \dots, 9_g\}$$

$$T = \{0_g|1_g, \dots, |9_g, 0_p|1_p, \dots, |9_p, a_g|b_g, \dots, |z_g, a_p|b_p, \dots, |z_p\}$$

Rules Set 1		Rules Set 2	
S	$\rightarrow \langle AP \rangle \langle X \rangle$	S	$\rightarrow \langle AG \rangle \langle X \rangle$
AP	$\rightarrow \langle Pstr_A \rangle \langle AP \rangle \mid \langle Pstr_A \rangle$	AG	$\rightarrow \langle Gstr_A \rangle \langle AG \rangle \mid \langle Gstr_A \rangle$
X	$\rightarrow \langle Gstr_N \rangle \langle AP \rangle \mid \langle Gstr_N \rangle$	X	$\rightarrow \langle Pstr_N \rangle \langle AG \rangle \mid \langle Pstr_N \rangle$

$$Gstr_A \rightarrow a_g|b_g|c_g, \dots, |z_g$$

$$Gstr_N \rightarrow 0_g|1_g|2_g, \dots, |9_g$$

$$Pstr_A \rightarrow a_p|b_p|c_p, \dots, |z_p$$

$$Pstr_N \rightarrow 0_p|1_p|2_p, \dots, |9_p$$

there are two different rules sets presented in Context Free Grammar (CFG) Defs. 1 for the integration of gesture and posture modalities as follows:

1. *Posture Alphabets and Gesture Numbers:*

Description: This grammar accepts the finger-spelling posture alphabets and gesture number signs. From the CFG grammar rules, firstly it detects two posture alphabet signs and then the gesture number sign is recognized or it detects firstly the posture alphabet sign, then gesture number sign and is followed by another posture alphabet.

2. *Posture Numbers and Gesture Alphabets:*

Description: This grammar accepts the finger-spelling posture numbers and gesture alphabet signs. From the CFG grammar rules, firstly it detects two gesture alphabet signs and then the posture number sign is detected or it detects firstly the gesture alphabet sign, then posture number sign and finally the second gesture alphabet sign.

Different symbols can be formed in integration process depending upon the lexicon, selected from gesture alphabet set (i.e., Rules Set 1) or postures

alphabet set (i.e., Rules Set 2) as shown in Table 6.1. The contribution-weights computed through particle filter system whose threshold (T) is above 70% are selected for the fusion process and is written as:

$$(\alpha_{gstr} | \alpha_{pstr}) \geq T \quad (6.11)$$

The inference for gesture symbols starts after some frames because drawing gesture symbols takes some time instances. In contrast, posture system recognizes the symbol at every time instance because a single frame is sufficient to recognize ASL symbols. Integration is carried out when contribution-weights of gesture α_{gstr} and posture α_{pstr} signs satisfy the threshold (T) at any time instance. In this regard, different approaches are proposed for the fusion of different systems which includes AND/OR combination, majority voting, behavior knowledge method and weighted voting method [146]. However, we have used AND/OR combination for recognized gesture and posture symbols. Integration I is formulated as:

$$I = \langle \alpha_{gstr} \bullet R_{hmm} \rangle \cap \langle \alpha_{pstr} \bullet R_{svm} \rangle \quad (6.12)$$

$$I = \langle \alpha_{gstr}^i \bullet R_{hmm}^i; i = 1 \dots m \rangle \cap \langle \alpha_{pstr}^j \bullet R_{svm}^j; j = 1 \dots n \rangle \quad (6.13)$$

In our experiments, the combination of CFG rules yield the integration of gesture and posture recognition in which multiple posture symbols (i.e., described above as n) are combined with a gesture symbol (i.e., m) or when multiple gesture symbols (i.e., m) are combined with a posture symbol (i.e., n).

To make inferences of results from CFG (see. CFG Defs. 1 (Rule Set 1)), the possible derivation of posture results is $\langle Pstr_A \rangle$ followed by a gesture number $\langle Gstr_N \rangle$ whereas $\langle Pstr_A \rangle$ yields the last outcome in the integration or $\langle Pstr_A \rangle$ followed by another posture symbol whereas $\langle Gstr_N \rangle$ comes after that. The inference derived from CFG rules are as follows:

$$S \rightarrow \langle Pstr_A \rangle \langle Gstr_N \rangle \langle Pstr_A \rangle \mid \langle Pstr_A \rangle \langle Pstr_A \rangle \langle Gstr_N \rangle$$

The second Rule Set defined in CFG Defs. 1 with the derivation which results in gesture alphabet sign $\langle Gstr_A \rangle$ followed by a posture number $\langle Pstr_N \rangle$ whereas $\langle Gstr_A \rangle$ yields the last outcome or in a second scenario where $\langle Gstr_A \rangle$

is detected followed by another gesture symbols whereas $\langle Pstr_N \rangle$ comes at the last.

$$S \rightarrow \langle Gstr_A \rangle \langle Pstr_N \rangle \langle Gstr_A \rangle \mid \langle Gstr_A \rangle \langle Gstr_A \rangle \langle Pstr_N \rangle$$

Based on the detected outcomes, different interpretations are devised for integration process which includes:

1. *Interpretation:*

$\langle Gesture \Rightarrow Detected \rangle ; \langle Posture \Rightarrow Detected \rangle ; \langle Integration \Rightarrow Yes \rangle$

Description: The ideal case of integration, both gesture and posture systems recognize the symbol at any time instance.

2. *Interpretation:*

$\langle Gesture \Rightarrow NotDetected \rangle ; \langle Posture \Rightarrow Detected \rangle ; \langle Integration \Rightarrow No \rangle$

Description : Gesture system does not classify any symbol because HMM is not giving any classification result when gesture drawing process starts. In contrast, the posture system classifies the sign with the contribution-weights α_{pstr} above the threshold.

3. *Interpretation:*

$\langle Gesture \Rightarrow SemiDetected \rangle ; \langle Posture \Rightarrow Detected \rangle ; \langle Integration \Rightarrow Yes/No \rangle$

Description: There can be some predictions about gesture symbols dependent upon the inference from HMM states. In this case, gesture symbol is still incomplete and it gives a clue about user's intention while drawing the gesture symbol. Intentions are predicted only when contribution-weight α_{gstr} of gesture sign pass the threshold criterion.

4. *Interpretation:*

$\langle Gesture \Rightarrow NotDetected \rangle ; \langle Posture \Rightarrow NotDetected \rangle ; \langle Integration \Rightarrow No \rangle$

Description: No match has occurred from gesture and posture systems. In this way, the symbols are not present in the lexicon.

6.4 Experimental Results and Analysis

In the proposed approach, experimental setup involves the tasks of data acquisition, gesture and posture classification and particle filter based integration process which is then linked to CFG inference rules to generate “meaningful expressions”. The applicability of proposed approach is demonstrated on real-time example scenarios and presented that how the meaningful expressions are generated from the integration of these systems. As the domain of research fields (i.e., both HCI and computer vision) is very much context sensitive and application oriented, so only a few ASL datasets are available such as [147] which are designed for specific applications with non-flexible assumptions.

Based on aforementioned information, the dataset in the laboratory comprising of 6 subjects performing the gesture and posture signs (i.e., presented in Section 5.1.5 and Section 5.2.2) where the image sequences are captured by Kinect camera with 480×640 pixels image resolution. In the proposed approach, for the integration, interpretation and inferences modules, no training has been performed, and the testing examples are independent to classification process. By doing so, the applicability can be extended by designing the lexicon and CFG rules according to the scenario under observation.

The proposed concept of integration is tested on a real-time example scenario, for instance we have designed restaurant lexicon which reflects the functionality of food and drink order placement at counter. For this purpose, we have studied type of food and drink item in a menu (e.g., name of fruit, drinks, fast food, etc.). In this scenario, we have chosen 45 different fruits for this choice as shown in Table. 6.1 and make different (i.e., currently our system supports about 500 combinations) choices for the menu-order by combining recognized gestures and postures signs. For example, an order can be placed through the integration of the first and second/third alphabet of the fruit name from gesture recognition whereas detected ASL posture describes the quantity of desired fruit item as shown in Fig. 6.3 and Table. 6.1. Moreover, the order can also be placed by integrating the first and second/third alphabet of the fruit name from ASL posture recognition whereas the detected gesture describes the quantity of desired fruit item as presented in Fig. 6.4.

Fig. 6.3 and Fig. 6.4 present an interpretation based on the integration of gesture and posture recognition system. In Fig. 6.3, the posture system firstly recognizes alphabet ‘B’ and it is followed by alphabet ‘A’. With this

combination, the proposed system considers it as a banana fruit. However, gesture recognition system did not recognize any symbol during the initial frames. At *Fr.* 105, it detects the gesture symbol ‘7’ indicating the quantity of order. Moreover, from *Fr.* 98 to 105, gesture recognition system computes the probability of possible gestures which the user can draw depending on HMM states and most likely candidates for the gesture recognition. It selects the highest probability element and mark it the ‘best’ element for recognition. The possible gestural symbols detected by HMM is ‘2’ but its probability is less than ‘7’. Therefore, the order is completed using the CFG rules results in “Seven Banana Juices” (i.e., $\langle Rec_{pstr} = 'B' \rangle$, $\langle Rec_{pstr} = 'A' \rangle$, $\langle Rec_{gstr} = '7' \rangle$). The first two graphs in Fig. 6.3 present the quantized features for the gesture recognition with Bezier descriptor $N = 15$ where the left graph results in Gesture ‘-1’ and the right graph results in Gesture ‘7’ for the sample frames. It is followed by the statistical and geometrical graphs for the posture recognition. Finally, in the last graph, classification rate and weight-contributions of gesture and posture recognition is presented for the complete sequence. The recognition of gesture and posture system after applying the threshold has been used for the integration of these systems. Fig. 6.4 presents the second case scenario where the interpretation starts when user draws the posture symbol ‘3’. Moreover, in this sequence, the two gesture symbols are detected at *Fr.* 110 and *Fr.* 190 as Gesture ‘A’ and Gesture ‘P’ describes *Apple*. So, the complete order from the gesture and posture recognition system is $\langle Three\ Apple\ Juices \rangle$, thus ordering (i.e., $\langle Rec_{pstr} = '3' \rangle$, $\langle Rec_{gstr} = 'A' \rangle$, $\langle Rec_{gstr} = 'P' \rangle$). Gesture and posture recognition work optimally and recognize the signs correctly (See Graphs in Fig. 6.4). By changing the lexicon, the proposed approach can be used for other scenarios as well.

In the graph of Fig. 6.4, the classification outcome of SVM in frames 155 – 169 is between 40-50% and after that, the classification outcome increases to 90% which is in the proposed approach termed as ambiguous classification outcome. So, this has also been addressed using the integration scheme through particle filter where the current state t is linked with its previous state at frame $t - 1$ and therefore, the previous weights in particle filter effects on the current classification outcome. So, under the ambiguous behavior of SVM, we can handle and control the computation of contribution weights with a particle filter. We argue that when the recognition result itself is considered as contribution-weights, the process of integration suffers due to this ambiguous

behavior.

We have tested the proposed approach on restaurant lexicon database with the overall 98.6% inference accuracy. It is observed that the classification inaccuracies do not effect the performance due to particle filter based weight computation technique. One of the potential reasons is, the particle filter works on the principle of prediction and updation mechanism, therefore, the inference of meaningful expression is achieved successfully.

6.5 Summary and Conclusion

In this chapter, the main objective is to integrate, interpret and finally infer meaningful expressions from the gesture and posture modalities. It begins with the concept of integration followed by particle filter system for measuring the contribution weights for gesture and posture modalities. These weights are used to resolve the ambiguities occurred in gesture and posture modalities and are utilized for the successful integration, generating the possible interpretations and finally deriving the inferences. These inferences are generated by designing the CFG rules for the gesture and posture modalities. The experimental results show that the proposed integration approach for the gesture and posture modules has successfully resulted in extracting the meaningful expressions with 98.6% recognition rates which proves the significance of proposed approach.

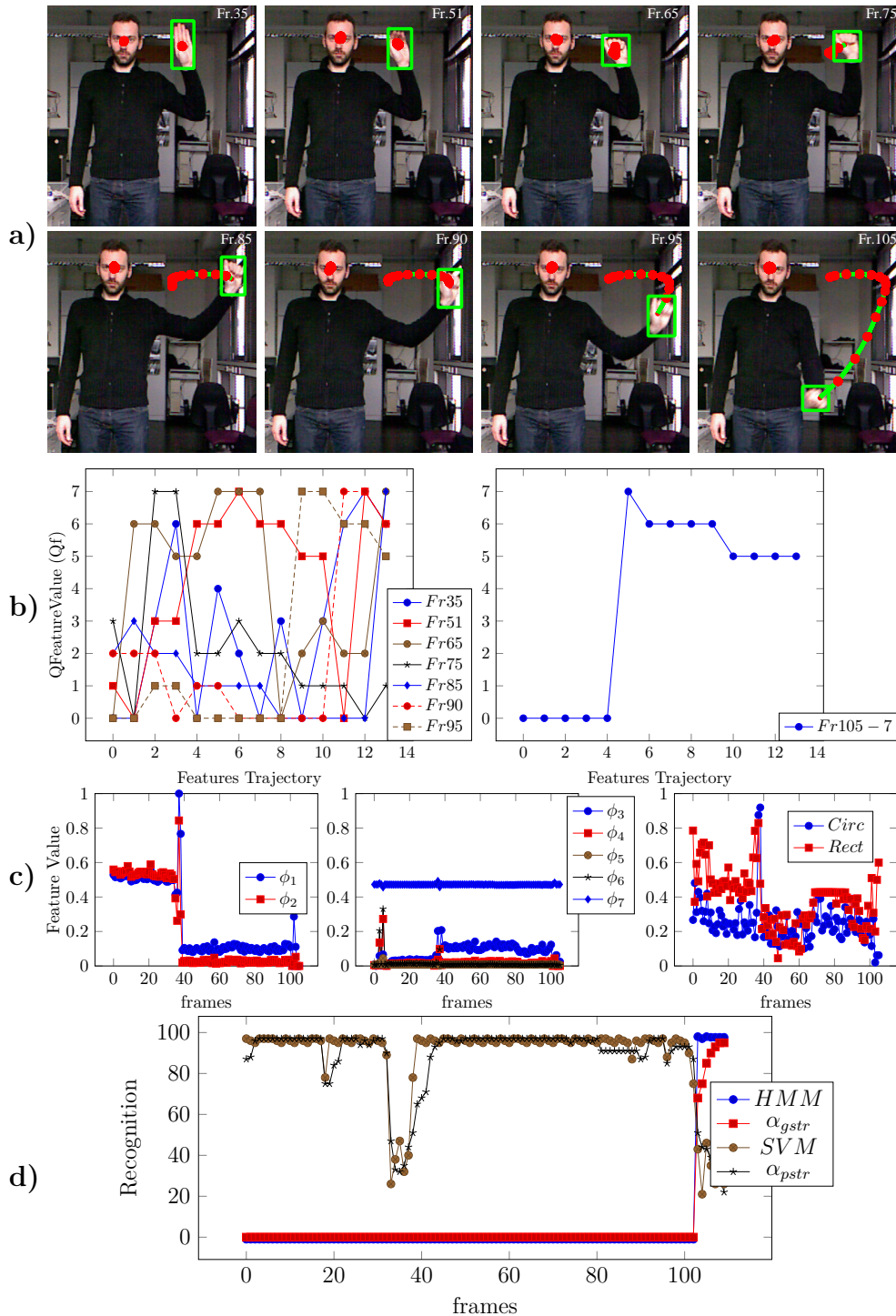


Figure 6.3: Meaningful expression “Seven Banana Juices” results from recognized ASL posture symbols ‘B’ followed by ‘A’ to result in ‘Banana’ and gesture ‘7’. a) presents the images from sequence 1. b) Gesture features from the Bezier descriptors c) Statistical and geometrical features for the posture recognition. d) Classification rate for the gesture and posture along with contribution-weights.

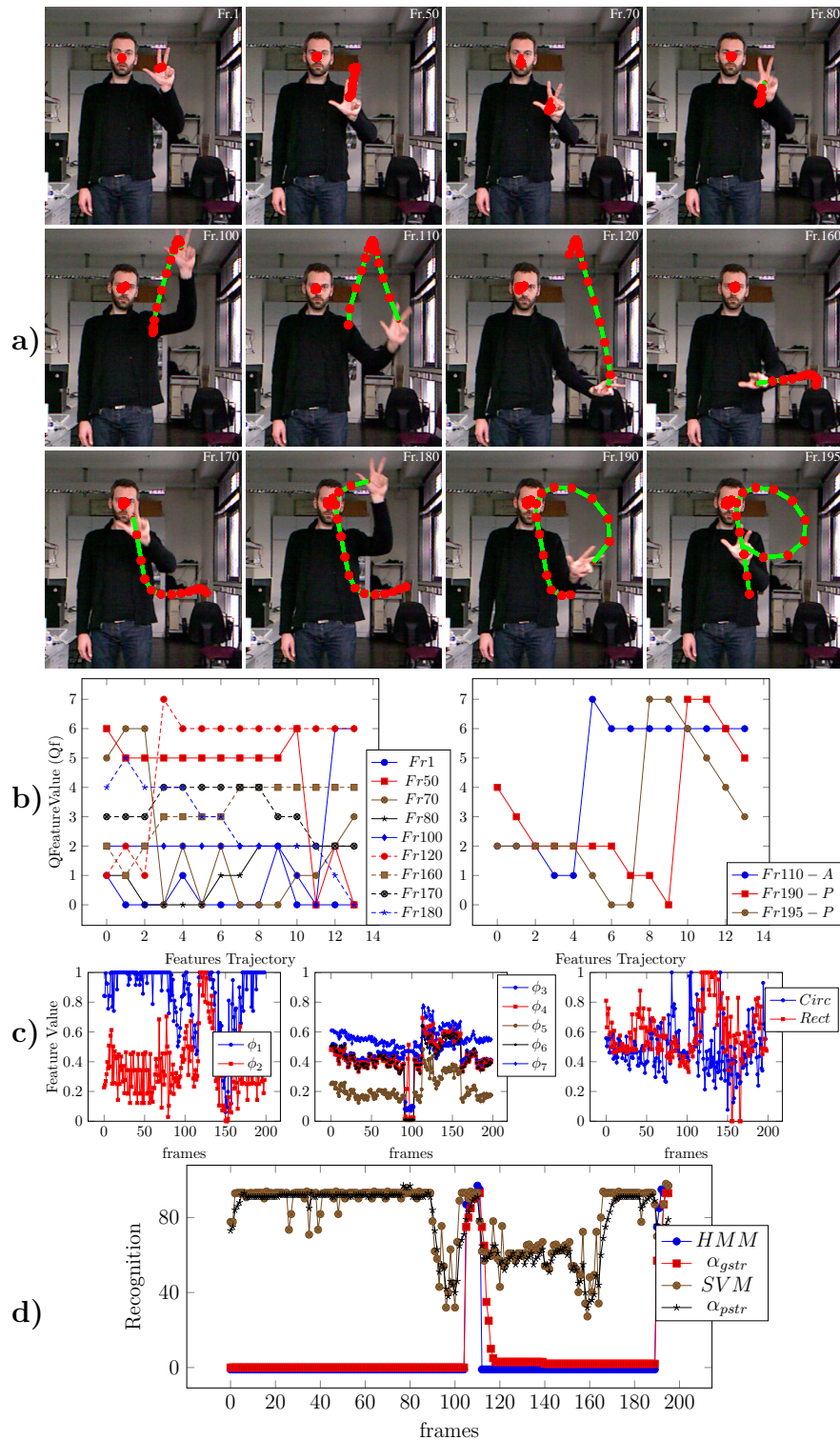


Figure 6.4: Meaningful expression “Three Apple Juices” results from recognized gesture symbols ‘A’ and ‘P’ and the posture symbol ‘3’. a) presents the images from sequence 2. b) Gesture features from the Bezier descriptors c) Statistical and geometrical features for the posture recognition. d) Classification rate for the gesture and posture along with contribution-weights.

Content Augmentation over Hand Postures

This chapter presents an extended concept of this work by augmenting the virtual contents over the hand postures. In the earlier chapters, we have described the proposed approaches for hand gesture and posture recognition and inference of meaningful expressions. Motivating with the fact of extending the applicability of this research (i.e., HCI as an assistant tool), the virtual components are augmented over the hand postures where the camera is adjusted in a tilted manner (i.e., 45° orientation) unlike the scenario in Section 5.1.5. This chapter begins with Section 7.1 which is dedicated to build the skeleton from the segmented hand. Based on the extracted features, Section 7.2 presents the process of determining the pose parameters of hand. Experimental results are demonstrated in Section 7.3 which shows the performance of proposed approach. Finally, this chapter ends with a summary and conclusion in Section 7.4.

7.1 Hand Skeleton Formation

In this section, the main aim is to build the skeleton model *Skelet* over hand posture. The proposed methodology (see Fig. 7.1) takes the input from the segmented hand (see Section 3.4) and determine the hand structure comprising of hand palm and fingers thus representing the hand geometry. Unlike the model based approaches in which the 3D kinematic hand model is built with certain Degrees of Freedom [26], the objective here to develop a real-time system using appearance based approach to offer flexible applicability in the domain of HCI.

The fingertips *FT* (see Section 4.3) and hand palm (see Section 3.4) are detected as separate components of the hand but no concrete inferences are derived from these isolated components. In the proposed approach, hand palm

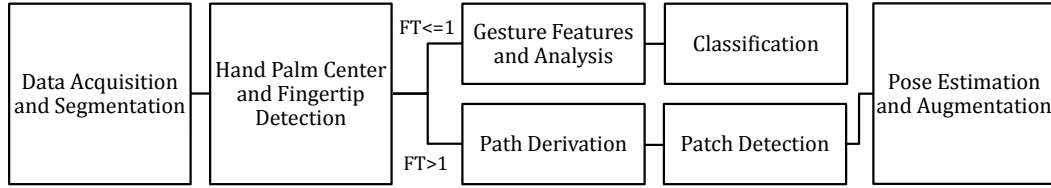


Figure 7.1: The framework for augmenting the virtual contents on hand postures.

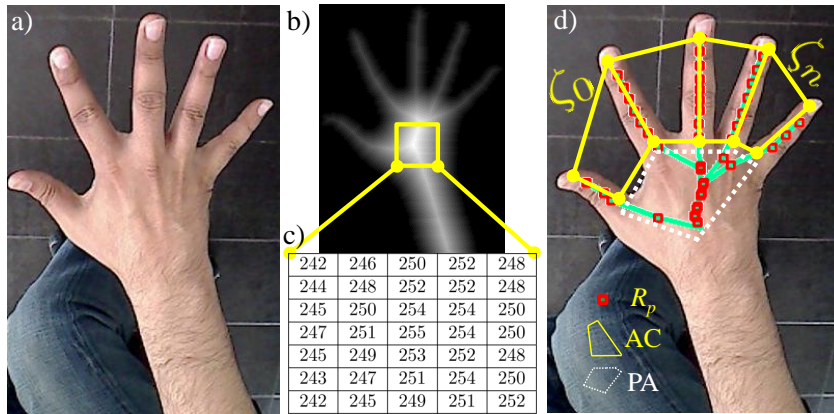


Figure 7.2: a) Original image b) Distance transformation of the image c) Distance transformation values d) Patch description ζ on the hand whereas R_p are the representative points from each finger to palm. AC and PA are the active (i.e., lies on the fingers) and passive (i.e., lies on the palm) representative points.

and fingertips f are utilized for building the hand skeleton model and the aim is to derive the association between the fingers and palm. The paths from fingertips to palm center is computed by incorporating the distance scores measured (see Section 3.4) and segmented skin pixels (see Section 3.2) within a search window. Practically, given the fingertip point, the traversing process is started by taking the search window of 3×3 for the path derivation process. In this window, the skin pixel value with the maximum distance score is selected and is marked as representative point (R_p). This process continues until R_p finds the optimal route to the palm's center. In this way, a list of R_p is obtained from *finger-to-palm* constituting the path $Path_{(f_v \rightarrow palm)}$. The same procedure is repeated for each finger f resulting in a structural representation like the actual hand physics as shown in Fig. 7.2 and Fig. 7.3.

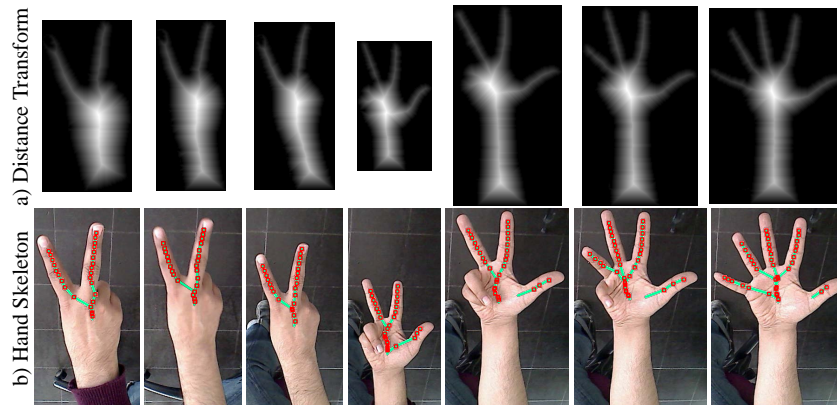


Figure 7.3: a) The distance transformation results for different hand poses in which the brighter values indicate the higher scores and vice versa. These scores help us to find the hand’s palm and by doing so, pruning the arm region. b) The path derivation process to draw the path from fingertips to palm center (i.e., shown by cyan path line with red key points) using the search region.

7.2 Hand Posture Geometry

Hand posture represents the stable state and geometry of the hand which is further encoded to classify various symbols (i.e., ASL, actions, commands etc.). Unlike the application scenario in Section 3.1.1, for augmenting the virtual component, the fundamental requirement is to measure the stable hand geometry and the corresponding pose. Pose estimation is a challenging task especially when the hand is varying in its shape and appearance continuously. In the proposed approach, we introduced the key idea of computing the patches over the hand physical structure and utilize them to estimate the pose correctly. This methodology leads to the development of natural and flexible hand based HCI applications by applying the advanced trends (i.e., in Augmented Reality). In the following, patch detection and augmentation process are presented as:

7.2.1 Patch Detection

Due to the deformable structure of hand, it is important either to model the whole hand with some mathematical model for finding the poses or to use some features (i.e., extracted from optical flow, or interest point detectors) and track them over time for pose estimation [148]. But, the main disadvantage of

Table 7.1: Average processing-time in milliseconds (640×480) for different modules of the proposed approach

Modules	Processing Time ms
Segmentation	18.8
Hand Detection	2.1
Fingertip Detection	4.3
Hand Structure	0.8
Patch Detection	0.9
Pose Estimation & Augmentation	11.2
Total time in msec	38.1

modelling approaches is the requirement of training whereas for the tracking, it is very hard to get consistent features in the homogeneous skin region of hand which results in ambiguous poses. Therefore, in the proposed approach, we present the idea of computing the patches over the physical structure of hand. Mathematically, the patches (ζ) are the surface regions which are derived from two neighboring fingers (i.e., finger-to-palm points) as presented in Fig. 7.2 d). Given the hand geometrical structure (i.e., comprises of set of representative points denoted as R_p), it is required to separate these R_p of finger-to-palm path into two regions named as active (AC) and passive (PA) regions by taking the mean. Active regions ($AC \subset Path_{(f_v \rightarrow palm)}$) are the R_p occupied by the fingers whereas the passive region ($PA \subset Path_{(f_v \rightarrow palm)}$) are the R_p that falls on the hand's palm (i.e., outside the finger region). Further, the first and last representative points R_p in AC are selected to establish the patches as shown in Fig. 7.2. A patch consists of four R_p selected from two neighboring fingers and are represented as:

$$\zeta_{(i,i+1)} = \{AC_{(f_i^{start})}, AC_{(f_i^{end})}, AC_{(f_{i+1}^{start})}, AC_{(f_{i+1}^{end})}\} \quad (7.1)$$

where i and $i + 1$ are the finger indexes for creating the patch, $start$ and end are the first and last R_p in AC . As the hand has a non-planar structure and it is difficult to model any consistent geometry on it, therefore, we represent them in the form of patches. Once we define our patches, it is necessary to compute the pose for every patch which are then integrated with other patches for the final pose estimation.

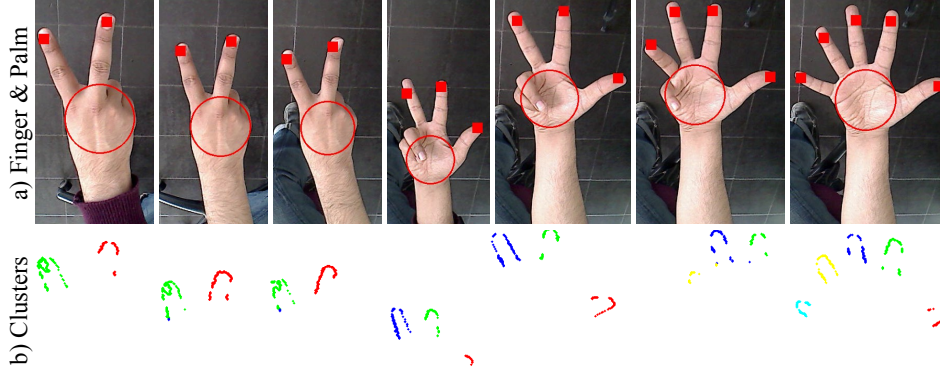


Figure 7.4: a) The palm (marked with red circle) and fingertip detections (marked with red filled rectangles) with the long and short sleeves in the image sequences. It is observed that the palm center point is un-effected by using the distance scores on the long or short sleeves. b) The clustered candidate regions for the detected fingertips.

7.2.2 Content Augmentation

The contents are augmented based on the computed pose over the hand postures. The pose comprises of translation $tr = \{t_x, t_y, t_z\}$ and rotation $ro = \{r_x, r_y, r_z\}$ parameters [149, 150]. These parameters are defined for every detected patch ζ as ζ^{tr} and ζ^{ro} which are then aggregated to form the final pose ξ . Mathematically, the detected patches for each hand skeleton is defined as:

$$Skelet : \zeta = \begin{cases} \zeta_i^{tr} \\ \zeta_i^{ro} \end{cases}, i = 1, 2, \dots, N \quad (7.2)$$

where ζ_i is a detected patch from the hand skeleton, ζ_i^{tr} are the translation parameters of a patch, ζ_i^{ro} are the rotation parameters and N is the total number of detected patches. To find these translation and rotation parameters of each patch, camera calibration is performed which gives the camera intrinsic parameters (i.e., principal focus and center points) and distortion parameters. Using these parameters and the patch representative points (R_p), the extrinsic parameters (i.e., translation and rotation parameters) are computed for each patch through $3D - 2D$ point correspondences using *solvePnP* algorithm [151]. The individual translation and rotation parameters of each patch are finally combined to get the final translation and rotation parameters of the

Table 7.2: Comparison between Fiducial Detection and Point Models (Mean Re-projection Error)

Patch Point Model	Average RMS Error (pixels)
4-Patch Points	3.7
8-Patch Points	2.3
16-Patch Points	2.3
Fiducial (Marker Detection)	0.45

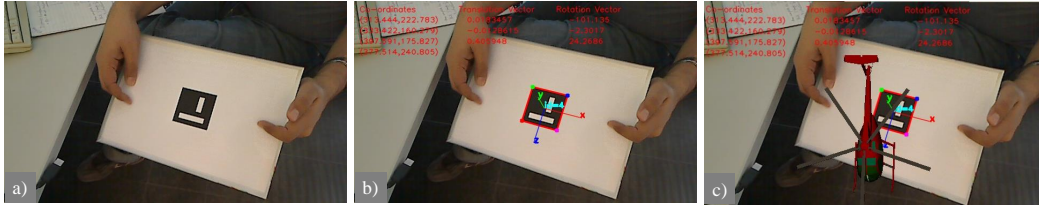


Figure 7.5: a) Original image with fiducial b) Detected pose over fiducial. c) Augmentation of 3D object over detected marker.

hand pose. It is defined as:

$$\xi = \begin{bmatrix} \sum_{i=1}^N \left(\frac{\zeta_i^{tr}}{N} \right) \\ \sum_{i=1}^N \left(\frac{\zeta_i^{ro}}{N} \right) \end{bmatrix} \quad (7.3)$$

Fig. 7.6 presents the individual patches information along with the final pose estimation. In addition, as the proposed approach presents an HCI application with AR system, so, the translation and rotation parameters of 3D objects are transformed by augmenting them on the hand posture as shown in Fig. 7.6 c).

7.3 Experimental Results and Analysis

The experiments are conducted in two different aspects, first, the virtual contents are overlaid over postures and second the gesture are classified (i.e., when there is only one fingertip or less). The experimental setup involves the tasks of data acquisition, skin color segmentation, hand skeleton formation, gesture classification, pose estimation and augmentation of 3D models. We



Figure 7.6: Results of content augmentation on image sequences. a) Original images b) Patch detection and camera pose estimation. c) Augmentation of 3D objects (i.e., kettle) on the hand posture.

have demonstrated the applicability of our proposed patch-based augmented reality system on real situations where the 3D models of different objects (i.e., aeroplanes, helicopters, kettles etc.) are augmented on the fly over the subject's hand postures satisfying the criteria of ease, flexibility and naturalness (i.e., with no clothing restriction).

The proposed framework runs with real-time processing at 25fps on Intel Processor 2.83GHz, 4 cores hardware configuration having 480×640 pixels image resolution. Table. 7.1 presents the processing time of each method along with the average processing time for each frame. The experiments are conducted on 50 video observations of four subjects performing various hand postures (i.e., with varying fingers) wearing short-to-long sleeves. Also, it is to be noted that our algorithm doesn't require any prior training for the hand skeleton as well as the pose estimation process for the augmentation.

In the experiments, we have used a low-cost web camera with two image resolutions 480×640 pixels and 240×320 based on the criterion of processing time versus image resolution. The processing time of images with resolution

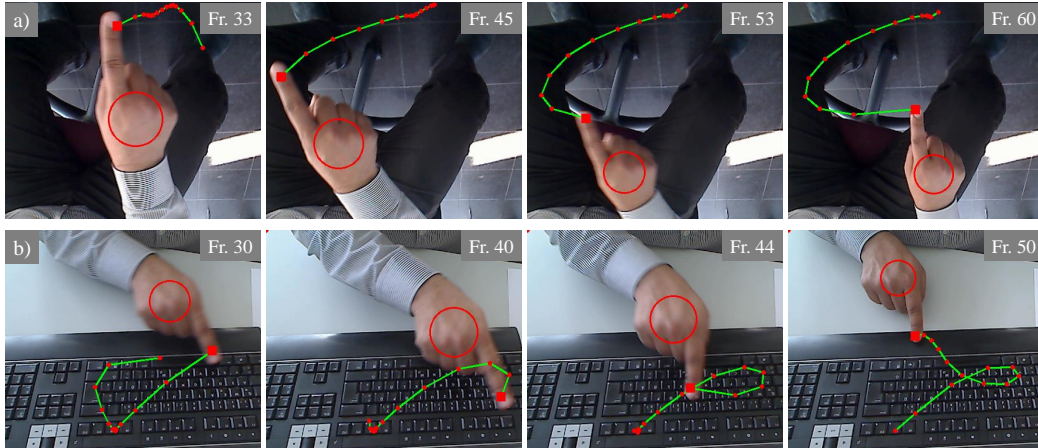


Figure 7.7: Bezier features extraction and classification results on image sequences with 1 fingertip detected. a) Image sequence for *Gesture* ‘C’ is presented at Fr 33 , Fr 45 , Fr 53 and Fr 60. HMM recognizes the symbol *Gesture* ‘C’ between Fr 53 and Fr 60. b) Image sequence for the gesture *Gesture* ‘K’ presented at Fr 30 , Fr 40 , Fr 44 and Fr 50. HMM recognizes the symbol *Gesture* ‘K’ between Fr 44 and Fr 50.

480×640 pixels is $25fps$ whereas the processing time of images with resolution 240×320 pixels is $48fps$. In this paper, we have selected the optimal image resolution for real-time processing (i.e., 480×640 pixels). However, we didn’t perform extensive tests on higher resolution (i.e., 960×1280) because in this case, the processing time increases which consequently leads to the decrease in frame rate (i.e., it works around $10 - 15fps$) and makes our application not suitable for real-time interactive scenarios. But, the proposed approach can be tested on GPU to optimize the performance for the high resolutions images, which is however, not the intended focus of this research.

In Fig. 7.6, the images of the sequence are presented in Fig. 7.6 a) whereas , Fig. 7.6 b) presents the formation of patches (ζ) between two detected fingers from representative points (R_p) on which the corresponding pose is estimated (i.e., marked by red regions). These individual poses are then aggregated to get the final pose of the hand posture. Finally, the pose estimation parameters (i.e., translation tr and rotation rot parameters) are transformed for 3D object to augment them on the hand postures as presented in Fig. 7.6 c). Moreover, the quantitative analysis is performed on our dataset for the pose estimation and augmentation where re-projection error is 2.3 pixels for *8-patch point model*. Table. 7.2 presents the comparison of different patch point models

where 8 – *point* and 16 – *point* models have the same mean re-projection error in pixels. Also, we have made a comparison with marker-based fiducial detection approach using Hamming distances for the pose estimation and augment the virtual object over the detected marker as shown in Fig. 7.5.

Fig. 7.7 shows the image sequence along with the feature extracted when one fingertip is detected. In the cases where none or one fingertip is detected, the proposed approach utilizes the detected fingertip for gesture features extraction. In the sequence for gesture recognition, the features are extracted using Bezier descriptors (i.e., $N = 15$) and HMM is used to recognize the gesture symbols. In Fig. 7.7 a), the image sequence for *Gesture* ‘C’ is presented at Fr 33 , Fr 45 , Fr 53 and Fr 60 where HMM recognizes the symbol *Gesture* ‘C’ between Fr 53 and Fr 60. Fig. 7.7 b) presents the second sequence for *Gesture* ‘K’ presented at Fr 30 , Fr 40 , Fr 44 and Fr 50. HMM recognizes the symbol *Gesture* ‘K’ between Fr 44 and Fr 50. In the experimental results like in Section 5.1.5, we have compared different Bezier descriptors and the original control points and observed that the performance of Bezier descriptor $N = 15$ is superior amongst all with recognition rate of 97.1%.

7.4 Summary and Conclusion

In this chapter, a hand skeleton approach is proposed to detect the physical components of the hand and their associated relationships (i.e., fingertip to palm connectivity). Over the computed skeleton, the hand pose is estimated by incorporating the suggested idea of patches, computed between the detected neighboring fingers. The individual extracted pose parameters are finally aggregated to augment different 3D objects on the hand. The experimental results are presented on IIKT-AR dataset to show the performance of the proposed approach. Moreover, the comparative analysis is carried out on different patch models and marker-based fiducial detection approach where the re-projection error is measured for the estimated pose.

Summary and Future Directions

In this thesis, the aim is to understand and detect the meaningful activities based on the hand gesture and posture modalities in HCI domain. This thesis is compiled into eight chapters, each addressing varied range of objectives and are linked in progressive manner allowing to develop the conceptual and practical understanding of the conducted research.

8.1 Summary

Chapter 1 is the front face of this thesis which begins with the motivation behind this research, provides the concept definition and application scenarios, identified the objectives and finally present the contributions. This chapter provides the theoretical and technical description of the research objectives and ends with the thesis structure.

Chapter 2 presents a detailed insight of relevant literature for gesture and posture modality along with point by point discussion. Besides, this chapter covers the literature of hand augmentation for mapping the virtual contents over the hand postures and pinpoints the key issues. Comprehensively, this chapter filters out the key research gaps which has been addressed during the entire course of this research.

Chapter 3 presents the hand and face segmentation problem which is an essential requirement for feature extraction and classification. The core idea is to use skin segmentation using Normal Gaussian distribution to get the raw classified skin information. The limitation of these methods has been addressed by suggesting a new approach that takes into account the detected face from Haar-like features which adaptively updates the skin segmentation process. As a result, this approach is scalable and functions optimally even when the non-trained data (i.e., different ethnicity, different lighting) is given

to the segmentation process. The blobs are extracted from the segmented objects (hand and face) and then distance transformation approach is incorporated to prune the detected arm for the subjects wearing half-sleeves (i.e., offer flexible conditions).

Chapter 4 addresses the topic of feature extraction for the gesture and posture modalities incorporating the global and local features respectively. In the gesture feature extraction process, the concept of Bezier curves has been employed to build Bezier descriptors from the hand centroid points. In addition, the fingertips are detected by computing the curvature on the extracted contours. These detected fingertips are used in the categorization process and to build the patches for the hand augmentation process. In the posture feature extraction, the statistical and geometrical features are employed along with categorization of hand fingers based on fingertip detection, separating into groups, thus leading to reduced mis-classifications significantly. Finally, the occlusion is handled through an iterative closest point algorithm which takes local features as the observation and resolve the ambiguities between the hands and face to maintain the tracking process.

Chapter 5 presents the classification scheme for the gesture and posture modalities and evaluates the performance of features. In the gesture recognition, HMM is employed for different Bezier descriptors along with the analysis based on the ground truth. In contrast, SVM is used for the posture classification by incorporating the statistical and geometrical features along with their analysis.

Chapter 6 presents the concept of integration of gesture and posture modalities to extract meaningful expression entailed from the designed logical models. A Particle filter is proposed to approximate the probability density function using a collection of random samples from classification outcome to generate the contribution-weights. These contribution-weights are used for the integration of gesture and posture modalities, the derived interpretation and computed inferences by incorporating CFG rules.

Chapter 7 extends the proposed framework of this thesis by augmenting the virtual contents over hand postures. A new approach is proposed to build

the hand skeleton by detecting and linking the hand physical components (i.e., using palm and fingertip detection). The hand skeleton is divided into patches computed from the detected neighboring fingers and camera poses are estimated for each patch. These estimated poses are then aggregated to generate final pose over which 3D objects are augmented. Finally, the quantitative analysis is performed by taking different patch point models and fiducial markers. Moreover, the fingertip and hand palm is used for gestural actions with the motivation to test the applicability of this research for different context.

8.2 Future Directions

The research is a self-evolving process and the objectives addressed in this thesis opens up new directions for future works. In the following, a summary of the future directions is presented within the context of gesture, posture and hand augmentation. Some of these key findings are described as follows:

- In the gesture recognition, dataset currently consisting of alphabets and numbers which can be extended into various directions (i.e., words, actions, commands etc.). Moreover, the gesture spotting is an important research field where the start and end of gestural symbols can be determined by building a model.
- The posture dataset consists of finger-spelling ASL alphabets and numbers which can be broadened to words as well. Moreover, a key research of ASL lies in the incorporation of facial feature in which the face features are extracted to infer the actions.
- The performance of occlusion process in tracking framework can be measured and improved by incorporating the hand shape or motion features which is fused to enhance the robustness for various contexts.
- The integration of gesture and posture leads to extract the meaningful expressions for the interpretations and inferences which will be extended for other application scenarios in HCI for determining the intention of the user.
- In the hand-based augmentation domain, currently, it is dependent upon the features extracted from the hand palm and fingertips. However, the

future research directions considers the self-occluded poses (i.e., where the fingers intercept each other while moving) of the hand-based augmentation.

APPENDIX A

Appendix

A.1 Orthogonal Moments

The absolute moments are formed using the monomial basic set $x^p y^q$. This property is for the cartesian moments and it is in non-orthogonal moment set. As it belongs to the non-orthogonal set, a high correlation exists between the moments. Due to the existence of high correlation, it needs high computational precision. Also, for the geometrical moments (i.e., absolute and central moments), it is hard to differentiate between different patterns because $x^p y^q$ powers are not very different from one another [152]. Teague in [153] proposed two inverse moment transformation techniques and how an image is reconstructed from set of moments. The first approach derives a continuous function moments exactly match the moments m_{pq} of $f(x, y)$ through order n . It is defined as:

$$f(x, y) = f_{00} + f_{10}x + f_{01}y + f_{20}x^2 + f_{11}xy + f_{02}y^2 + \dots \quad (\text{A.1})$$

This approach was not suitable when higher moments are used due to the complexities of the equations in it. The second method used is based on orthogonal moments. Teague examines that the Cartesian moment can be replaced by orthogonal basic set (i.e. Legendre and Zernike polynomial), resulting in an orthogonal moment set. The basic advantage of using the orthogonal moments is that they can be represented with the differences to the same accuracy as the monomial set with low computational precision. The orthogonal moments are computed in a similar manner as geometrical moments. The difference is that the monomials are replaced by the set of polynomials. The orthogonal moment is written in the similar manner as the geometrical moment and is defined as:

$$m_{pq} = \iint_R h_{pq}(x, y) f(x, y) dx dy \quad (\text{A.2})$$

where $h_{pq}(x, y)$ is pq -th orthogonal polynomial and R is the range over this polynomial is defined. The Legendre polynomial is defined as:

$$h_{pq}(x, y) = L_p(x - \bar{x}) L_q(y - \bar{y}), \quad -1 \leq x, y \leq 1 \quad (\text{A.3})$$

The Zernike polynomial is defined as:

$$h_{pq}(x, y) = Z_{pq}(x - \bar{x}, y - \bar{y}), \quad x^2 + y^2 \leq 1 \quad (\text{A.4})$$

Legendre moments are translation and scale invariant but not rotation invariant. Zernike moments are translation, scale and rotation invariant. These moments are used in reconstruction of images as they have got the advantage of inverse moment transformation.

A.1.1 Zernike Moments:

Teague [153] examines that the Cartesian moment can be replaced by orthogonal basic set (i.e. Zernike polynomial), resulting in an orthogonal moment set. The magnitudes of Zernike moments are invariant to rotation and reflection [154]. However, translation and scaling invariance can easily be achieved like the central moments.

$$A_{pq} = \frac{(p+1)}{\Pi} \sum_x \sum_y I(x, y) [V_{pq}(x, y)], \quad x^2 + y^2 \leq 1 \quad (\text{A.5})$$

where $I(x, y)$ is image pixel and p and q defines the moment-order. Zernike polynomials $V_{pq}(x, y)$ are defined in polar form $V_{pq}(r, \theta)$ as:

$$V_{pq}(r, \theta) = R_{pq}(r) e^{-jq\theta} \quad (\text{A.6})$$

where R_{pq} is a radial polynomial and is defined as:

$$R_{pq}(r) = \sum_{s=0}^{\frac{p-|q|}{2}} \frac{(-1)^s (p-s)! r^{p-2s}}{s! \left(\frac{p+|q|}{2} - s\right)! \left(\frac{p-|q|}{2} - s\right)!} \quad (\text{A.7})$$

We have used the Zernike moments upto 4th order moment. The feature vector set for Zernike moment is as under:

$$F_{Zernike} = \{Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8, Z_9\} \quad (\text{A.8})$$

Normalization: The normalization is done for features to keep them in a particular range and is defined as:

$$c_{min} = \mu - 2\sigma, \quad c_{max} = \mu + 2\sigma \quad (\text{A.9})$$

$$nF_i = (F_i - c_{min}) / (c_{max} - c_{min}) \quad (\text{A.10})$$

$nF_{Zernike}$ are the normalized features for Zernike. c_{max} and c_{min} are the respective maximum and minimum values used for the normalization.

A.2 Experimental Setup



Figure A.1: a) First Context (IIKT-GP): Kinect camera is oriented in front of the subject for hand gesture and posture recognition. b) Second Context (IIKT-AR), webcam is adjusted in front of the subject in a tilted manner (i.e., 45 orientation) for AR scenario.

Bibliography

- [1] Allan Pease and Barbara Pease. *The Definitive Book of Body Language*. London Orion Books Ltd., 2005. (Cited on page 1.)
- [2] Nicu Sebe, Ira Cohen, Theo Gevers, and Thomas S. Huang. Multimodal approaches for emotion recognition: a survey. *Internet Imaging VI*, 5670:56–67, 2005. (Cited on page 1.)
- [3] M. Pantic and M.S. Bartlett. Machine analysis of facial expressions. In K. Delac and M. Grgic, editors, *Face Recognition*, pages 377–416. I-Tech Education and Publishing, Vienna, Austria, July 2007. (Cited on page 1.)
- [4] Fengyi Song, Xiaoyang Tan, Songcan Chen, and Zhi-Hua Zhou. A literature survey on robust and efficient eye localization in real-life scenarios. *Pattern Recognition*, 46(12):3157–3173, 2013. (Cited on page 1.)
- [5] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304, Washington, DC, USA, 2011. IEEE Computer Society. (Cited on pages 1 and 28.)
- [6] Sebastian Handrich and Ayoub Al-Hamadi. A robust method for human pose estimation based on geodesic distance features. In *IEEE International Conference on Systems, Man, and Cybernetics, Manchester, SMC 2013, United Kingdom, October 13-16, 2013*, pages 906–911, 2013. (Cited on page 1.)
- [7] X. Zabulis, H. Baltzakis, and A.A. Argyros. *Vision-based Hand Gesture Recognition for Human-Computer Interaction*. Human Factors and Ergonomics. 2009. (Cited on page 1.)
- [8] Ankit Chaudhary, J.L. Raheja, Karen Das, and Sonia Raheja. A survey on hand gesture recognition in context of soft computing. In Natarajan Meghanathan, BrajeshKumar Kaushik, and Dhinakaran Nagamalai,

- editors, *Advanced Computing*, volume 133 of *Communications in Computer and Information Science*, pages 46–55. Springer Berlin Heidelberg, 2011. (Cited on page 1.)
- [9] S de la Rosa, S Mieskes, HH Bülthoff, and C Curio. View dependencies in the visual recognition of social interactions. *Frontiers in Psychology*, 4(752):1–10, 10 2013. (Cited on page 1.)
- [10] R. Sathya and M. Kalaiselvi Geetha. Article: Vision based traffic police hand signal recognition in surveillance video - a survey. *International Journal of Computer Applications*, 81(9):1–10, November 2013. Full text available. (Cited on page 1.)
- [11] H. Yeo, B. Lee, and H. Lim. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. *Multimedia Tools and Applications*, pages 1–29, 2013. (Cited on pages 1, 14 and 45.)
- [12] Xu Zhang, Xiang Chen, Wen-hui Wang, Ji-hai Yang, Vuokko Lantz, and Kong-qiao Wang. Hand gesture recognition and virtual game control based on 3d accelerometer and emg sensors. In *Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI '09*, pages 401–406, New York, NY, USA, 2009. ACM. (Cited on page 1.)
- [13] Sang-Heon Lee, Myoung-Kyu Sohn, Dong-Ju Kim, Byungmin Kim, and Hyunduk Kim. Smart tv interaction system using face and hand gesture recognition. In *Consumer Electronics (ICCE), 2013 IEEE International Conference on*, pages 173–174, Jan 2013. (Cited on page 1.)
- [14] Brad A. Myers. A brief history of human-computer interaction technology. *interactions*, 5(2):44–54, March 1998. (Cited on page 2.)
- [15] Alejandro Jaimes and Nicu Sebe. Multimodal human-computer interaction: A survey. *Comput. Vis. Image Underst.*, 108(1-2):116–134, October 2007. (Cited on page 2.)
- [16] Omer Rashid, Ayoub Al-Hamadi, and Bernd Michaelis. Utilizing invariant descriptors for finger spelling american sign language using svm. In *6th international conference on Advances in visual computing - Volume Part I, ISVC'10*, pages 253–263, 2010. (Cited on pages 2, 28, 43 and 58.)

-
- [17] Leyla Norooz, Matthew L Mauriello, Anita Jorgensen, Brenna McNally, and Jon E Froehlich. Bodyvis: A new approach to body learning through wearable sensing and visualization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1025–1034. ACM, 2015. (Cited on page 2.)
- [18] D. Dimov, M. Alexander, and N. Zlateva. Cbir approach to the recognition of a sign language alphabet. In *International Conference on Computer Systems and Technologies CompSysTech 07*, pages 1–9, 2007. (Cited on page 2.)
- [19] Farid Parvini, Dennis Mcleod, Cyrus Shahabi, Bahareh Navai, Baharak Zali, and Shahram Ghandeharizadeh. An approach to glove-based gesture recognition. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques*, pages 236–245, Berlin, Heidelberg, 2009. Springer-Verlag. (Cited on page 2.)
- [20] P. Maes P. Mistry. Sixthsense Ū a wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches*, 2009. (Cited on page 3.)
- [21] J. David Sturman and David Zeltzer. A survey of glove-based input. *IEEE Computer Graphics Applications*, 14(1):30–39, Jan 1994. (Cited on page 3.)
- [22] Farid Parvini, Dennis Mcleod, Cyrus Shahabi, Bahareh Navai, Baharak Zali, and Shahram Ghandeharizadeh. An approach to glove-based gesture recognition. In *13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques*, pages 236–245, Berlin, Heidelberg, 2009. Springer-Verlag. (Cited on page 3.)
- [23] L. Lamberti and F. Camastra. Real-time hand gesture recognition using a color glove. In *Image Analysis and Processing*, pages 365–373. Springer, 2011. (Cited on page 3.)
- [24] S. Ahmad. A usable real-time 3d hand tracker. In *Signals, Systems and Computers*, pages 1257–1261, 1994. (Cited on page 11.)
- [25] John Lin, Ying Wu, and T. S. Huang. Modeling the constraints of human hand motion. In *Proceedings of the Workshop on Human Motion*

- (*HUMO'00*), HUMO '00, pages 121–126, Washington, DC, USA, 2000. IEEE Computer Society. (Cited on page 11.)
- [26] Pragati Garg, Naveen Aggarwal, and Sanjeev Sofat. Vision based hand gesture recognition. 3(1):821 – 826, 2009. (Cited on pages 11 and 109.)
- [27] J. Appenrodt, S. Handrich, A. Al-Hamadi, and B. Michaelis. Multi stereo camera data fusion for fingertip detection in gesture recognition systems. In *Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of*, pages 35–40, Dec 2010. (Cited on page 11.)
- [28] Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:677–695, 1997. (Cited on pages 11, 28 and 29.)
- [29] SiddharthS. Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, pages 1–54, 2012. (Cited on pages 11 and 12.)
- [30] Saikat Basak and Arundhuti Chowdhury. Article: A vision interface framework for intuitive gesture recognition using color based blob detection. *International Journal of Computer Applications*, 90(15):36–40, March 2014. Full text available. (Cited on pages 12 and 34.)
- [31] Prashan Premaratne. *Human Computer Interaction Using Hand Gestures*. Springer, Singapore, 2014. (Cited on pages 12 and 34.)
- [32] Xiaohui Shen, Gang Hua, Lance Williams, and Ying Wu. Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields. *Image Vision Comput.*, 30(3):227–235, March 2012. (Cited on page 12.)
- [33] Ming hsuan Yang, Narendra Ahuja, and Mark Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *PAMI*, 24:1061–1074, 2002. (Cited on pages 12 and 45.)
- [34] Ahmet Birdal and Reza Hassanpour. Region based hand gesture recognition. In *16th International conference in central Europe on computer graphics, visualization and computer vision*. Václav Skala-UNION Agency, 2008. (Cited on page 12.)

- [35] SX. Ju, M.J. Black, S. Minneman, and Kimber D. Analysis of gesture and action in technical talks for video indexing. Technical report, American Association for Artificial Intelligence. AAAI Technical Report SS-97-03, 1997. (Cited on page 12.)
- [36] L. Bretzner, I. Laptev, and T. Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Automatic Face and Gesture Recognition, Proceedings of Fifth IEEE International Conference, 2002*, pages 423–428, 2002. (Cited on pages 13 and 17.)
- [37] A. Bellarbi, H. Belghit, S. Benbelkacem, N. Zenati, and M. Belhocine. Hand gesture interaction using color-based method for tabletop interfaces. In *7th International Symposium on Intelligent Signal Processing (WISP 11), IEEE*, pages 180–185, 2011. (Cited on pages 14 and 17.)
- [38] Q. Chen, N. Georganas, and E. Petriu. Real-time vision based hand gesture recognition using haar-like features. In *Instrumentation and Measurement Technology Conference Proceedings*, pages 1–6, 2007. (Cited on pages 14 and 17.)
- [39] Mahmoud Elmezain, Ayoub Al-Hamadi, Jörg Appenrodt, and Bernd Michaelis. A hidden markov model-based isolated and meaningful hand gesture recognition. 2(5):1083 – 1090, 2008. (Cited on pages 14, 17 and 45.)
- [40] Chung-Lin Huang and Sheng-Hung Jeng. A model-based hand gesture recognition system. *Mach. Vis. Appl.*, 12(5):243–258, 2001. (Cited on pages 14 and 17.)
- [41] Hong Li and Michael A. Greenspan. Multi-scale gesture recognition from time-varying contours. In *ICCV*, pages 236–243. IEEE Computer Society, 2005. (Cited on pages 14 and 17.)
- [42] Y. Wen, C. Hu, G. Yu, and C. Wang. A robust method of detecting hand gestures using depth sensors. In *IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE)*, pages 72–77, 2012. (Cited on pages 15 and 17.)

-
- [43] H. Nanda and K. Fujimura. Visual tracking using depth data, sep 2009. US Patent 7,590,262. (Cited on pages 15 and 17.)
- [44] Antonis A. Argyros and Manolis I. A. Lourakis. Binocular hand tracking and reconstruction based on 2d shape matching. In *ICPR (1)*, pages 207–210. IEEE Computer Society, 2006. (Cited on pages 15 and 17.)
- [45] Michael Van den Bergh and Luc J. Van Gool. Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *WACV*, pages 66–72. IEEE Computer Society, 2011. (Cited on pages 15 and 17.)
- [46] Cheoljong Yang, Yujeong Jang, Jounghoon Beh, David Han, and Hanseok Ko. Gesture recognition using depth-based hand tracking for contactless controller application. In *IEEE International Conference on In Consumer Electronics (ICCE)*, pages 297–298, 2012. (Cited on pages 16 and 17.)
- [47] Jagdish L. Raheja, Ankit Chaudhary, and Kunal Singal. Tracking of fingertips and centers of palm using kinect. *Computational Intelligence, Modelling and Simulation, International Conference on*, 0:248–252, 2011. (Cited on pages 16 and 17.)
- [48] Jounghoon Beh, David K. Han, Ramani Durasiwami, and Hanseok Ko. Hidden markov model on a unit hypersphere space for gesture trajectory recognition. *Pattern Recognition Letters*, 36:144–153, 2014. (Cited on pages 16 and 17.)
- [49] Qing Chen. *Real-time Vision-based Hand Tracking and Gesture Recognition*. PhD thesis, Ottawa, Ont., Canada, Canada, 2008. AAINR41629. (Cited on page 17.)
- [50] Richard Bowden, Andrew Zisserman, Timor Kadir, and Mike Brady. Vision based interpretation of natural sign languages. In *In: Exhibition at ICVS03: The 3rd International Conference on Computer Vision Systems*, pages 391–401. ACM Press, 2003. (Cited on page 18.)
- [51] Liwei Zhao, Karin Kipper, William Schuler, Christian Vogler, Norman I. Badler, and Martha Stone Palmer. A machine translation system from

- english to american sign language. In John S. White, editor, *AMTA*, volume 1934 of *Lecture Notes in Computer Science*, pages 54–67. Springer, 2000. (Cited on page 18.)
- [52] W. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1995. (Cited on pages 18 and 22.)
- [53] M. Handouyahia, D. Ziou, and S. Wang. Sign language recognition using moment-based size functions. In *International Conference of Vision Interface*, pages 210–216, 1999. (Cited on pages 18 and 22.)
- [54] P. Frosini. Measuring shapes by size functions. 1607:122–133, 1991. (Cited on page 19.)
- [55] Omnia S. ElSaadany and Moataz M. Abdelwahab. Real-time 2dhog-2dpca algorithm for hand gesture recognition. In Alfredo Petrosino, editor, *ICIAP (2)*, volume 8157 of *Lecture Notes in Computer Science*, pages 601–610. Springer, 2013. (Cited on pages 19, 22 and 45.)
- [56] S. Liwicki and M. Everingham. Automatic recognition of fingerspelled words in british sign language. In *2nd IEEE Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 09), in conjunction with CVPR2009*, pages 50–57, Los Alamitos, CA, USA, 2009. IEEE Computer Society. (Cited on pages 19, 22, 34, 37 and 42.)
- [57] J. Wu and W. Gao. The recognition of finger-spelling for chinese sign language. In *Gesture and Sign Language in Human-Computer Interaction*, pages 96–100. Springer-Verlag, 2002, 2002. (Cited on pages 19, 22 and 34.)
- [58] J. Isaacs and S. Foo. Hand pose estimation for american sign language recognition. In *36th Southeastern Symp. System Theory*, Lecture Notes in Computer Science, pages 132–136, 2004. (Cited on page 19.)
- [59] V. Ayala-Ramirez, S. A. Mota-Gutierrez, U. H. Hernandez-Belmonte, and R. E. Sanchez-Yanez. A hand gesture recognition system based on geometric features and color information for human computer interaction tasks. In *ROSSUM 2011*, 2011. (Cited on pages 19, 22, 34 and 42.)

- [60] Nicolas Pugeault and Richard Bowdven. Spelling it out: Real-time asl fingerspelling recognition. In *ICCV Workshops*, pages 1114–1119. IEEE, 2011. (Cited on pages 20 and 22.)
- [61] A. Braffort. Argo: An architecture for sign language recognition and interpretation. In *Proceedings of Gesture Workshop on Progress in Gestural Interaction*, pages 17–30. Springer-Verlag, 1997, 1997. (Cited on pages 20, 22 and 34.)
- [62] M. Zahedi, D. Keysers, and H. Ney. Appearance-based recognition of words in american sign language. In *Iberian Conf. on Pattern Recognition and Image Analysis*, pages 511–519. Springer-Verlag, 2005, 2005. (Cited on pages 20 and 22.)
- [63] César Roberto de Souza and Ednaldo Brigante Pizzolato. Sign language recognition with support vector machines and hidden conditional random fields: Going from fingerspelling to natural articulated words. In *Proceedings of the 9th International Conference on Machine Learning and Data Mining in Pattern Recognition*, MLDM’13, pages 84–98, Berlin, Heidelberg, 2013. Springer-Verlag. (Cited on page 20.)
- [64] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001. (Cited on page 20.)
- [65] Gary R. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Proceedings Fourth IEEE Workshop on Applications of Computer Vision, WACV 1998, October 19-21, 1998, Princeton, New Jersey, USA*, pages 214–219, 1998. (Cited on page 20.)
- [66] N. P. Vassilia and G. M. Konstantinos. On feature extraction and sign recognition for greek sign language. In *International Conference on Artificial Intelligence and Soft Computer*, pages 93–98, 2003. (Cited on pages 20 and 22.)
- [67] M. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(No. 8):1061–1074, 2002. (Cited on pages 20 and 22.)

- [68] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 759–760, New York, NY, USA, 2011. ACM. (Cited on pages 21 and 22.)
- [69] A. Licsar. Supervised training based hand gesture recognition system. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)*, pages 210–216. IEEE Computer Society, 2002, 2002. (Cited on pages 21, 22 and 42.)
- [70] S. Malassiotis and M. Srinivasan. Real-time hand posture recognition using range data. 26:1027–1037, 2008. (Cited on pages 21 and 22.)
- [71] Songfan Yang and B. Bhanu. Understanding discrete facial expressions in video using an emotion avatar image. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(4):980–992, 2012. (Cited on page 23.)
- [72] Brendan Klare, Roman V. Yampolskiy, and Anil K. Jain. Face recognition in the virtual world: recognizing avatar faces. In *In Proc. of SPIE, Biometric Technology for Human Identification IX*, 2012. (Cited on page 23.)
- [73] Justin N Oursler, Mathew Price, Roman V Yampolskiy, and JB Speed Hall. Parameterized generation of avatar face dataset. In *14th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games*, pages 17–22, 2009. (Cited on page 23.)
- [74] Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1&A2):52 – 73, 2007. (Cited on page 23.)
- [75] J. Chun and S. Lee. A vision-based 3d hand interaction for marker-based ar. *International Journal of Multimedia and Ubiquitous Engineering*, 7(3):51–58, 2012. (Cited on pages 23, 24, 25 and 26.)

- [76] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, August 1997. (Cited on page 23.)
- [77] T. Noell, A. Pagani, and D. Stricker. Markerless camera pose estimation - an overview. In *Visualization of Large and Unstructured Data Sets - Applications in Geospatial Planning, Modeling and Engineering*, volume 19 of 4, pages 45–54, 2011. (Cited on page 23.)
- [78] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn. Fingartips: Gesture based direct manipulation in augmented reality. In *2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia (GRAPHITE 04)*, pages 212–221, 2004. (Cited on page 23.)
- [79] T. Lee and T. Hoellerer. Hybrid feature tracking and user interaction for markerless augmented reality. In *IEEE Virtual Reality*, pages 145–152, 2008. (Cited on pages 23, 24 and 26.)
- [80] S. Malassiotis and M.G. Strintzis. Real-time hand posture recognition using range data. *Image and Vision Computing*, 26(7):1027–1037, 2008. (Cited on page 23.)
- [81] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, CVPR '05, pages 590–596, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on pages 24 and 25.)
- [82] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2Nd IEEE and ACM International Workshop on Augmented Reality, IWAR '99*, pages 85–, Washington, DC, USA, 1999. IEEE Computer Society. (Cited on pages 24 and 25.)
- [83] Youngkwan Cho, Jongweon Lee, and Ulrich Neumann. A multi-ring color fiducial system and an intensity-invariant detection method for scalable fiducial-tracking augmented reality. In *In IWAR*, pages 147–165, 1998. (Cited on page 24.)

- [84] T. Lee and T. Hoellerer. Handy ar: Markerless inspection of augmented reality objects using fingertip tracking. In *ISWC 2007*, pages 83–90, 2007. (Cited on pages 24 and 26.)
- [85] K. P. Ng, G. Y. Tan, and Y. P. Wong. Vision-based hand detection for registration of virtual objects in augmented reality. *International Journal of Future Computer and Communication*, 2(5):423–427, 2013. (Cited on pages 25 and 26.)
- [86] J. C. Chun and B. S. Lee. Dynamic manipulation of a virtual object in marker-less ar system based on both human hands. *Transactions on Internet and Information Systems*, 4(4):618–632, 2010. (Cited on pages 25 and 26.)
- [87] B. S. Lee and J. C. Chun. Interactive manipulation of augmented objects in marker-less ar using vision-based hand mouse. In *International Conference on Information Technology(ITNG)*, pages 398–403, 2010. (Cited on pages 25 and 26.)
- [88] R. Radkowski and C. Stritzke. Interactive hand gesture-based assembly for augmented reality applications. In *Proceedings of the 2012 International Conference on Advances in Computer-Human Interactions*, pages 303–308. IEEE, 2012. (Cited on pages 25 and 26.)
- [89] Sukeshini A. Grandhi, Gina Joue, and Irene Mittelberg. Understanding naturalness and intuitiveness in gesture production: Insights for touch-less gestural interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 821–824, New York, NY, USA, 2011. ACM. (Cited on pages 28 and 29.)
- [90] Sergios Theodoridis and Konstantinos Koutroumbas. Chapter 2 - classifiers based on bayes decision theory. In Sergios Theodoridis and Konstantinos Koutroumbas, editors, *Pattern Recognition (Fourth Edition)*, pages 13 – 89. Academic Press, Boston, fourth edition edition, 2009. (Cited on page 30.)
- [91] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001. (Cited on page 31.)

- [92] Guojun Lu. Chain code-based shape representation and similarity measure. In Clement Leung, editor, *Visual Information Systems*, volume 1306 of *Lecture Notes in Computer Science*, pages 135–150. Springer Berlin Heidelberg, 1997. (Cited on page 31.)
- [93] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 511–518, 2001. (Cited on page 32.)
- [94] Ricardo Fabbri, Luciano Da F. Costa, Julio C. Torelli, and Odemir M. Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.*, 40(1):2:1–2:44, February 2008. (Cited on page 35.)
- [95] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993. (Cited on page 35.)
- [96] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, 1988. (Cited on page 35.)
- [97] Kuo-Chin Fan, Den-Fong Chen, and Ming-Gang Wen. Skeletonization of binary images with nonuniform width via block decomposition and contour vector matching. *Pattern Recognition*, 31(7):823 – 838, 1998. (Cited on page 35.)
- [98] J. H. Conway, N. J. A. Sloane, and E. Bannai. *Sphere-packings, Lattices, and Groups*. Springer-Verlag New York, Inc., New York, NY, USA, 1987. (Cited on page 35.)
- [99] Gunilla Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371, 1986. (Cited on pages 36 and 37.)
- [100] S. Chu and J. Tanaka. Hand gesture for taking self portrait. In *Proceedings of the 14th international conference on Human-computer interac-*

- tion: interaction techniques and environments - Volume Part II*, HCII 11, pages 238–247, 2011. (Cited on pages 37 and 42.)
- [101] Fredrik Andersson. Bezier and B-Spline Technology. Technical report, Umea Universitet, Sweden, 2003. (Cited on page 45.)
- [102] Dianne Hansford. Chapter 4 - bézier techniques. In Gerald Farin, Josef Hoschek, and Myung-Soo Kim, editors, *Handbook of Computer Aided Geometric Design*, pages 75 – 109. North-Holland, Amsterdam, 2002. (Cited on page 45.)
- [103] Rida T. Farouki. The bernstein polynomial basis: A centennial retrospective. *Comput. Aided Geom. Des.*, 29(6):379–419, August 2012. (Cited on pages 45 and 46.)
- [104] David Salomon. *Curves and Surfaces for Computer Graphics*. Springer-Verlag New York, 2006. (Cited on page 46.)
- [105] M. Hu. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, 8(2):179–187, 1962. (Cited on page 55.)
- [106] S. Maitra. Moment invariants. In *IEEE Conf. on CVPR*, volume 67, pages 697–699, 1979. (Cited on page 55.)
- [107] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Journal of Pattern Recognition*, 26(1):164–174, 1993. (Cited on page 55.)
- [108] J. Davis and G. Bradski. Real-time motion template gradients using intel cvlib. In *IEEE ICCV Workshop on Framerate Vision*, 1999. (Cited on page 55.)
- [109] R. Prokop and A. Reeves. A survey of moment based techniques for un-occluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, 54:438–460, 1992. (Cited on page 57.)
- [110] A. Ross and R. Govindarajan. Feature Level Fusion Using Hand and Face Biometrics. In *Proceedings of SPIE*, pages 196–204, 2005. (Cited on page 61.)
- [111] Paul Smith, Niels da Vitoria Lobo, and Mubarak Shah. Resolving hand over face occlusion. In Nicu Sebe, Michael Lew, and ThomasS. Huang,

- editors, *Computer Vision in Human-Computer Interaction*, volume 3766 of *Lecture Notes in Computer Science*, pages 160–169. Springer Berlin Heidelberg, 2005. (Cited on page 63.)
- [112] Matilde Gonzalez, Christophe Collet, and Rémi Dubot. Head tracking and hand segmentation during hand over face occlusion in sign language. In Kiriakos N. Kutulakos, editor, *Trends and Topics in Computer Vision*, volume 6553 of *Lecture Notes in Computer Science*, pages 234–243. Springer Berlin Heidelberg, 2012. (Cited on page 63.)
- [113] Thomas Coogan, George Awad, Junwei Han, and Alistair Sutherland. Real time hand gesture recognition including hand segmentation and tracking. In George Bebis, Richard Boyle, Bahram Parvin, Darko Kopracin, Paolo Remagnino, Ara Nefian, Gopi Meenakshisundaram, Valerio Pascucci, Jiri Zara, Jose Molineros, Holger Theisel, and Tom Malzbender, editors, *Advances in Visual Computing*, volume 4291 of *Lecture Notes in Computer Science*, pages 495–504. Springer Berlin Heidelberg, 2006. (Cited on page 63.)
- [114] Alexander Ladikos, Selim Benhimane, and Nassir Navab. A real-time tracking system combining template-based and feature-based approaches. In *IN VISAPP*, 2007. (Cited on page 63.)
- [115] Feng-Sheng Chen, Chih-Ming Fu, and Chung-Lin Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 21(8):745 – 758, 2003. (Cited on page 63.)
- [116] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09*, pages 331–340. INSTICC Press, 2009. (Cited on page 64.)
- [117] C. Silpa-Anan and R. Hartley. Optimised kd-trees for fast image descriptor matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. (Cited on page 64.)
- [118] L. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, Vol. 77, No. 2, pages 257–286, 1989. (Cited on page 68.)

- [119] Pierre A. Devijver. *Hidden Markov Models*. Number C2 in Tutorial/11th IAPR Conference on Pattern Recognition. Technische Universiteit Delft, 1992. Applications in Pattern Recognition and Real-Time Modeling of Image Sequences. (Cited on page 68.)
- [120] L. Rabiner and B.H. Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, Jan 1986. (Cited on page 68.)
- [121] Jr. Vasko, R.C., A. El-Jaroudi, and J.R. Boston. An algorithm to determine hidden markov model topology. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 6, pages 3577–3580 vol. 6, May 1996. (Cited on page 70.)
- [122] X. D. Huang, Y. Ariki, and M. Jack. *Hidden Markov Models for Speech Recognition*. *Edinburgh University Press*, 1990. (Cited on page 71.)
- [123] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden markov models. *Mach. Learn.*, 29(2-3):245–273, November 1997. (Cited on pages 72, 73 and 74.)
- [124] G. D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE, Vol. 61*, pages 168–278, 1973. (Cited on page 74.)
- [125] A. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory, Vol. 13, No. 2*, pages 260–269, 1967. (Cited on page 74.)
- [126] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973. (Cited on page 74.)
- [127] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics, Vol. 41, No. 1*, pages 164–171, 1970. (Cited on pages 75 and 76.)
- [128] M. Soriano, S. Huovinen, B. Martinkauppi, and M. Laaksonen. Skin Detection in Video Under Changing Illumination Conditions. *In Proceeding International Conference on Pattern Recognition*, pages 839–842, 2000. (Cited on page 75.)

-
- [129] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000. (Cited on pages 82 and 83.)
- [130] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999. (Cited on page 83.)
- [131] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. (Cited on pages 83 and 84.)
- [132] C.J. Lin and R. Weng. Simple probabilistic predictions for support vector regression. Technical report, National Taiwan University, 2004. (Cited on page 84.)
- [133] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011. (Cited on page 84.)
- [134] Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014. (Cited on page 88.)
- [135] Lindsay I Smith. A tutorial on principal components analysis. Technical report, Cornell University, USA, February 26 2002. (Cited on page 89.)
- [136] A. Ross and A. Jain. Multimodal biometrics: An overview. In *Proceedings of 12th European Signal Processing Conference*, pages 1221–1224, 2004. (Cited on page 95.)
- [137] Roberto Brunelli and Daniele Falavigna. Person identification using multiple cues. *IEEE Trans. on PAMI*, 17:955–966, 1995. (Cited on page 95.)
- [138] Kyong Chang, Kevin W. Bowyer, and Patrick J. Flynn. Face recognition using 2d and 3d facial data. In *ACM Workshop on Multimodal User Authentication*, pages 25–32, 2003. (Cited on page 95.)
- [139] Qiang Wu, Liang Wang, Xin Geng, Ming Li, and Xiangjiang He. Dynamic biometrics fusion at feature level for video-based human recognition. pages 152–157, 2007. (Cited on page 95.)

- [140] Yunhong Wang, Tieniu Tan, and AnilK. Jain. Combining face and iris biometrics for identity verification. In Josef Kittler and MarkS. Nixon, editors, *Audio- and Video-Based Biometric Person Authentication*, volume 2688 of *Lecture Notes in Computer Science*, pages 805–813. Springer Berlin Heidelberg, 2003. (Cited on page 95.)
- [141] Robert W. Frischholz and Ulrich Dieckmann. Bioid: A multimodal biometric identification system. *Computer*, 33(2):64–68, February 2000. (Cited on page 95.)
- [142] Tanzeem Choudhury, Brian Clarkson, Tony Jebara, and Alex Pentland. Multimodal person recognition using unconstrained audio and video. In *2nd International Conference on Audio and Video-Based Boemtric Person Authentication*, pages 176–181, 1999. (Cited on page 95.)
- [143] Ajay Kumar, David Wong, Helen Shen, and Anil Jain. Personal verification using palmprint and hand geometry biometric. In *4th Int. Conf. on Audio and Video-based Biometric Person Authentication*, pages 668–678, 2003. (Cited on page 96.)
- [144] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *Int. Jour. of Computer Vision*, 29:5–28, 1998. (Cited on page 96.)
- [145] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Pearson Addison-Wesley, 2. edition, 2001. (Cited on page 100.)
- [146] Md. Monwar and Marina Gavrilova. A robust authentication system using multiple biometrics. In *Comp. and Information Science*, pages 189–201. 2008. (Cited on page 102.)
- [147] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Yuan Q., and A. Thangali. The asl lexicon video dataset. In *CVPR, Workshop on Human Communicative Behaviour Analysis*, 2008. (Cited on page 104.)
- [148] Martin de La Gorce, David J. Fleet, and Nikos Paragios. Model-based 3d hand pose estimation from monocular video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(9):1793–1805, September 2011. (Cited on page 111.)

-
- [149] George Stockman and Linda G. Shapiro. *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001. (Cited on page 113.)
- [150] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. (Cited on page 113.)
- [151] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. (Cited on page 113.)
- [152] M. Pawlak. *Image analysis by moments: reconstruction and computational aspects*. Oficyna Wydawn. Politechn., 2006. (Cited on page 122.)
- [153] Michael Reed Teague. Image analysis via the general theory of moments*. *J. Opt. Soc. Am.*, 70(8):920–930, Aug 1980. (Cited on pages 122 and 123.)
- [154] R.R. Bailey and M. Srinath. Orthogonal moment features for use with parametric and non-parametric classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(4):389–399, Apr 1996. (Cited on page 123.)

List of Publications

The presented thesis has the following international peer-reviewed journals and conference papers:

Journal Publications and Book Chapters

- S. Handrich, O. Rashid and A. Al-Hamadi, “Non-intrusive Gesture Recognition in Real Companion Environments”, SFB Book - Companion Technology, Book Chapter [Submitted].
- F. Saxen, O. Rashid, A. Al-Hamadi, S. Adler, A. Kernchen, R. Mecke, “Image-Based Methods for Interaction with Head-Worn Worker-Assistance Systems”, Journal of Intelligent Learning Systems and Applications, Aug. 2014, vol. 6, 141-152.
- O. Rashid, A. Al-Hamadi and K. Dietmayer, “Interpretation of Meaningful Expressions by Integrating Gesture and Posture Modalities”, International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM 2012), ISSN: 2012-7988, Vol. 4: 589-597, 2012.
- O. Rashid and A. Al-Hamadi, “An Integrated HCI Framework for Interpreting Meaningful Expressions”, International Journal of Computer Issues (IJCSI2012), ISSN: 1694-0814, Vol. 9(5), 411-421, Sep 2012.
- S.S. Pathan, O. Rashid, A. Al-Hamadi, B. Michaelis, “Multi-Object Tracking in Dynamic Scenes By Integrating Statistical and Cognitive Approaches”, International Journal of Computer Science Issues (IJCSI 2012), ISSN: 1694-0814, Vol. 9 (4), 180-189, July 2012.
- A. Al-Hamadi, O. Rashid, and B. Michaelis, “Posture Recognition Using Combined Statistical and Geometrical Feature Vectors based on SVM”, International Journal of Information and Mathematical Sciences, ISSN: 2010-4065, 6(1): 7-14, 2010.
- M. Elmezain, A. Al-Hamadi, O. Rashid, B. Michaelis, “Posture and Gesture Recognition for Human-Computer Interaction”, In-tech, Advanced Technologies, ISBN 978-953-307-009-4, Book Chapter, 2009.

Conference Papers:

- O. Rashid, A. Al-Hamadi: “Utilizing Bezier Descriptors for Hand Gesture Recognition”, International Conference on Image Processing (ICIP 2015), Sep 27-30, 2015.
- O. Rashid, A. Al-Hamadi: “A New Approach for Hand Augmentation Based on Patch Modelling”, Advanced Concepts for Intelligent Vision Systems (ACIVS 2013),162-171, Oct 28-31, 2013.
- J. Tümler, A. Kernchen, R. Mecke, F. Saxen, O. Rashid, A. Al-Hamadi, A. Köpsel, A. Huckauf, “Companion-basiertes Assistenzsystem für die Mitarbeiterschulung in der Automobilindustrie”. Fachtagung Digital Engineering, 2013.
- F. Saxen, O. Rashid, A. Al-Hamadi, S. Adler, A. Kernchen, R. Mecke, “Image-Based Gesture Recognition for User Interaction with Mobile Companion-based Assistance Systems”, 4th International Conference of Soft Computing and Pattern Recognition (SoCPaR 2012), University Brunei Darussalam, Brunei, December 10-13, 2012.
- S. Handrich, O. Rashid, A. Al-Hamadi, “Improvement of Gesture Recognition Using Multi-hypotheses Object Association”, International Conference on Image and Signal Processing (ICISP 2012), Agadir Morocco, 28-30 Jun, 2012.
- O. Rashid and A. Al-Hamadi, “Flow Modeling and Skin-based Gaussian Pruning to Recognize Gestural Actions using HMM”, International Conference on Pattern Recognition (ICPR 2012), pp. 3488-3491, November 11-15 , Tsukuba Science City, Japan, 2012.
- O. Rashid and A. Al-Hamadi, “Recognizing Gestural Actions”, IEEE International Conference on Systems, Man, and Cybernetics, (SMC 2012), pp. 2682-2686, October 14-17, 2012, COEX, Seoul, Korea.
- O. Rashid, A. Al-Hamadi, B. Michaelis, “Robust Hand Posture Recognition with Micro and Macro Level Features using Kinect”, IEEE 3rd International Conference on Intelligent Computing and Intelligent Systems (ICIS 2011), Guangzhou China, 18-20 Nov, 2011.
- S. S. Pathan, A. Al-Hamadi, O. Rashid, B. Michaelis, “Learning A-priori Threshold to Initialize Flow-based Adaptive Mixture Model for Dynamic Scene Segmentation”, IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2011), Guangzhou China, Nov 18-20, 691-695, 2011.

- O. Rashid, A. Al-Hamadi, B. Michaelis, “Interpreting Dynamic Meanings by Integrating Gesture and Posture Recognition System”, 2nd International Workshop on Video Event Categorization, Tagging and Retrieval (VECTaR 2010), in ACCV 2010, Queenstown, 8-12 Nov, 2010.
- O. Rashid, A. Al-Hamadi, B. Michaelis, “Utilizing Invariant Descriptors for Finger Spelling American Sign Language using SVM”, 6th International Symposium on Visual Computing (ISVC 2010), Las Vegas, Nov 29 - Dec 1, 2010.
- O. Rashid, A. Al-Hamadi, B. Michaelis, “Integration of Gesture and Posture Recognition Systems for Interpreting Dynamic Meanings using Particle Filter”, International Conference of Soft Computing and Pattern Recognition (SoCPaR2010), Paris, Nov 7 - 10, 2010.
- O. Rashid, A. Al-Hamadi, B. Michaelis, “A Framework for the Integration of Gesture and Posture Recognition Using HMM and SVM”, IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2009), Shanghai China, 20-22 Nov, 2009.
- O. Rashid, A. Al-Hamadi, B. Michaelis, “Posture Recognition using Combined Statistical and Geometrical Feature Vectors based on SVM”, International Conference on Image, Signal and Vision Computing (ICISVC 2009), WASET, Singapore, 26-28 Aug, 2009.

Curriculum Vitae

Name	Omer Rashid Ahmad
Date of Birth	Mar 20, 1983 in Riyadh, Saudi Arabia
Nationality	Pakistani
Status	Married
Address	Weinbergstr. 47, 39106 Magdeburg
Email	omer.ahmad@ovgu.de
Education	<p>Jan 2001 - Apr 2005 Bachelor in Computer Engineering , University of Engineering and Technology, Lahore Pakistan</p> <p>Oct 2006 - May 2009 Master of Computational Visualistics, Otto-von-Guericke University Magdeburg, Germany</p> <p>Jun 2009 - present PhD Research, IIKT, Otto-von-Guericke University Magdeburg, Germany</p>
Professional Experience	<p>Position: Software Engineer</p> <p>Jul 2006 - Sep 2007, Streaming Networks Pvt. Ltd, Islamabad, Pakistan</p>

Magdeburg, Apr 27, 2015
Omer Rashid Ahmad

