



Hand Gesture Spotting and Recognition Using HMMs and CRFs in Color Image Sequences

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

von **M.Sc. Mahmoud Othman Selim Mahmoud Elmezain**

geb. am 08. December 1973 in Menofiya, Ägypten

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik
der Otto-von-Guericke-Universität Magdeburg

Gutachter:

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

Prof. Dr.-Ing. habil. Bernd Michaelis

Prof. Dr. Aly Farag

Promotionskolloquium am: 26. November 2010

This work is dedicated to ...
my parents, my wife (Rabab) and my children (Salma, Sara and Omnia)

Mahmoud

Abstract

Even though automatic hand gesture recognition technology has been applied to real-world applications with relative success, there are still several problems which need to be addressed for wider applications of Human Computer Interaction (HCI). One of such problems which arise in hand gesture recognition is to extract (spot) meaningful gestures from the continuous sequence of the hand motions. Another problem is caused by the fact that there is quite a bit of variability (i.e. in shape, trajectory and duration) in the same gesture even for the same person. Throughout literature, the backward spotting technique is used which first detects the end points of gestures and then tracks back through their optimal paths to discover the start points of gestures. Upon the detection of the start and the end points, in between points trajectory is sent to the recognizer for recognition. So, a time delay is observed between the meaningful gesture spotting and recognition. This time delay is unacceptable for online applications. Given the fact of high variability of corresponding gesture to other gestures, modeling the other gesture patterns (i.e. non-gesture patterns are other movements which do not correspond to gestures) is a vital issue to accommodate the infinite number of non-gesture patterns.

In this thesis, a forward gesture spotting system is proposed which handles hand gesture spotting and recognition simultaneously in stereo color image sequences without time delay. In addition, color and depth map which is obtained by passive stereo measuring based on the mean absolute difference and the known calibration data of the camera, are used to localize hands. Moreover, the hand trajectory is obtained by using Mean-shift algorithm in conjunction with depth map. This structure correctly extracts a set of hand postures to track the hand motion and achieves accurate and robust hand tracking with a stereo camera as an input device. One of the main contributions in the work is to examine the capabilities of combined features of location, orientation and velocity for gesture recognition with respect to Cartesian and Polar coordinates. Furthermore, k -means clustering algorithm is used to quantize the extracted features and employs them for Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) codewords. The effectiveness of these features yields reasonable recognition rates.

In this work, isolated gestures are handled according to two different classification

techniques: generative model such as HMMs and discriminative models like CRFs, Hidden Conditional Random Fields (HCRFs) and Latent-Dynamic Conditional Random Fields (LDCRFs) to decide the best in terms of recognition results. To spot meaningful gestures accurately, a stochastic method for designing a non-gesture model with HMMs versus CRFs is proposed with no training data. The non-gesture model provides a confidence measure which is used as an adaptive threshold to find the start and the end points of meaningful gestures which are embedded in the input video stream. The number of states of non-gesture model with HMMs increases as the number of gesture models increases. However, an increase in the number of states is nothing but lead to a waste of time and space. To alleviate this problem, a relative entropy which merges similar probability distribution states is used in order to save time, space, and to increase the spotting speed. On the other hand, the non-gesture model with CRFs is improved by adding a short gesture detector to further increase gestures spotting accuracy and also tolerate errors caused by spatio-temporal variabilities.

Another contribution is to use a forward spotting scheme in conjunction with sliding window mechanism to handle hand gesture segmentation and recognition at the same time. In addition, it solves the issues of time delay between meaningful gesture spotting and recognition and achieves accurate, robust results, as well as making the system capable of working for real-time applications.

To demonstrate coaction of the suggested components and the effectiveness of gesture spotting and recognition system, an application of gesture-based interaction with alphabets and numbers is implemented. The HMMs models are trained by Baum-Welch (BW) algorithm while CRFs are trained using gradient ascent along Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization technique. The experiments demonstrate that the proposed systems with HMMs and CRFs are accurate and efficient for spatio-temporal variabilities. In addition, these systems automatically recognize isolated and meaningful hand gestures with superior performance and low computational complexity when applied to several video samples containing complex situations.

Zusammenfassung

Obwohl eine Technologie zur Handgestenerkennung bereits mit relativ großem Erfolg in Realworld-Applikationen Verwendung findet, existieren immer noch einige Probleme die für tiefgreifendene Anwendungen im Bereich der Mensch-Computer-Interaktion (HCI), gelöst werden müssen. Eines dieser Probleme, welches sich im Bereich der Gestenerkennung aufgetan hat, ist die zuverlässige Extraktion bedeutungsreicher Gesten aus kontinuierlichen Bildsequenzen. Ein anderes Problem besteht in der Varianz (bezüglich Form, der Bahn, d.h. des zeitlichen Positionsverlaufs des erfassten Ziels und Dauer der Bewegung) von Gesten, sogar wenn diese von einer Person stammen. In der Literatur wird stets die „backward spotting“ Technik angegeben, bei welcher zunächst die Endpunkte einer Geste detektiert und anschließend deren optimaler Pfad verfolgt wird, um den Anfangspunkt der Geste zu ermitteln. Nachdem Anfangs- und Endpunkt bestimmt sind, werden die dazwischen befindlichen Punkte des Gestenpfades an den Klassifikator zur Erkennung weitergeleitet. In diesem Zusammenhang wurde eine Verzögerung zwischen Beobachtung und der Erkennung der bedeutungsreichen Gesten beobachtet. Diese zeitliche Verzögerung ist für online-Anwendungen inakzeptabel. Aufgrund der hohen Korrespondenz zwischen unterschiedlichen Gesten ist es wichtig für diese ein Muster zu entwerfen, um sich an die unendliche Anzahl von nicht-Gesten anzupassen.

In dieser Arbeit wird ein vorwärts gerichtetes Gestenerkennungssystem vorgestellt, welches Handgestenverfolgung und Erkennung in Sequenzen von Stereo-Farbbildern gleichzeitig und ohne zeitliche Verzögerung behandelt. Zusätzlich werden Farb- und Tiefenkarten benutzt - welche durch passive Stereo-Messungen, basierend auf der mittleren absoluten Differenz und den bekannten Kamerakalibrierungen berechnet werden - um die Hände zu lokalisieren. Der Verlauf der Handbewegung kann mit Hilfe des Meanshift-Algorithmus in Verbindung mit den Tiefenkarten berechnet werden. Diese Struktur extrahiert einen Satz von Handpositionen, mit welchen sich die Handbewegung verfolgen und mit Hilfe von Stereo-Kameras eine genaue und robuste Handverfolgung erreichen lässt. Einer der wesentlichen Beiträge dieser Arbeit ist es zu untersuchen, welche Möglichkeiten von kombinierten Merkmalen wie Position, Ausrichtung und Beschleunigung für eine Gestenerkennung hinsichtlich der Kartesischen und Polar-Koordinaten bestehen. Des Weiteren werden die extrahierten Merkmale

von k -means Algorithmen quantisiert und für Hidden Markov Modelle (HMMs) und Condition Random Fields (CRFs) eingesetzt. Die Effektivität dieser Merkmale kann akzeptable Erkennungsraten sicherstellen.

In dieser Arbeit werden isolierte Gesten von zwei verschiedenen Klassifikationstechniken behandelt; Erzeugungsmodelle wie HMMs und Unterscheidungsmodelle wie CRFs, Hidden Condition Random Fields (HCRFs) und latent-dynamischen CRFs, um entscheiden zu können, welcher Ausdruck das beste Ergebnis repräsentiert. Es wird eine stochastische Methode vorgeschlagen, die ohne Trainingsdaten nicht-Gesten Modelle mit HMMs bzw. CRFs erstellt, um bedeutungsreiche Gesten akkurat verfolgen zu können, wobei die Ergebnisse beider Klassifikatoren miteinander verglichen werden. Das nicht-Gesten Modell stellt dabei ein Konfidenzmaß bereit, das als adaptiver Schwellwert benutzt wird, um die Anfangs- und Endpunkte bedeutungsreicher Gesten zu finden. Die Anzahl der Zustände der nicht-Gesten Modelle verhält sich bei den HMMs proportional zur Anzahl der Gesten Modelle. Ferner ist eine Erhöhung der Anzahl von Zuständen lediglich Verschwendung von Zeit und Speicherplatz. Um die Anzahl von Zuständen zu reduzieren wird eine relative Entropie eingeführt und benutzt um ähnliche Wahrscheinlichkeitsverteilungen zu mischen, um dadurch Zeit und Speicherplatz zu sparen sowie die Geschwindigkeit der Zielverfolgung zu erhöhen. Andererseits wird das nicht-Gesten Modell mit CRFs verbessert, indem ein Kurzgesten Detektor hinzugefügt wird, um die Genauigkeit der Gestenerkennung weiter ansteigen zu lassen und Fehler tolerieren zu können, die durch raumzeitliche Variationen verursacht werden.

Ein weiterer Beitrag besteht darin vorwärts gerichtete Verfolgungsschemata in Verbindung mit sliding window Mechanismen zu benutzen, um eine Segmentierung und Erkennung von Handgesten zur gleichen Zeit betreiben zu können, was das Problem der Zeitverzögerung löst und das System akkurat und robust macht, so dass es sich für die Verwendung in Echtzeitapplikationen eignet.

Um das Zusammenspiel der vorgeschlagenen Komponenten und die Effektivität der Gestenverfolgung und Erkennung zu demonstrieren, wurde eine Anwendung zur gestenbasierten Interaktion mit Buchstaben und Nummern implementiert. Die HMMs wurden mit dem Baum-Welch (BW) Algorithmus trainiert, wogegen für CRFs ein Gradientenabstiegsverfahren nach der Optimierungstechnik von Broyden-Fletcher-Goldfarb-Shanno (BFGS) trainiert wurden. Die Experimente zeigen, dass die vorgeschlagenen Systeme mit HMMs und CRFs akkurat und effizient bezüglich raumzeitlicher Variationen arbeiten. Zudem vermag es das System isolierte und bedeutungsreiche Gesten mit überragender Performanz und geringer mathematischer Komplexität automatisch zu erkennen, wenn es auf verschiedene Videos mit komplexem Inhalt angewendet wird.

Acknowledgement

I would like to express my deep gratitude to everyone who helped me shape the ideas explored in this dissertation, either by giving technical advice or encouraging and supporting my work in many other ways. This dissertation would not have come into existence without their hands-on advice and motivation.

First of all, I am deeply indebted to my country “EGYPT” for accepting and supporting me to do my Ph.D. in Germany. I also have to thank my family, my father, my mother, my wife, and my kids for moral support, encouragement, and understanding.

I’m greatly indebted to my supervisor, Prof. Dr.-Ing. habil. Bernd Michaelis for being a consistent source of support and encouragement. His guidance and help have made my Ph.D. program a smooth and enjoyable one. I am extremely grateful to Prof. Dr.-Ing. habil. Ayoub Al-Hamadi, for guiding my work from the very first day and for supporting me in several directions. He gave me the opportunity to conduct this doctoral research and helped me made the right strategic decisions at many forks along the way. He kept me on track while allowing me to broaden my research horizon in tangential areas. His insightful comments, which densely filled the margins of each draft that I gave to him, gave rise to many creative ideas. I am deeply indebted to Prof. Dr. Aly A. Farag, University of Louisville, USA, for accepting to review my thesis. Also, thanks to Prof. U. Jumar and Prof. M. Leone for being on the examination committee.

I always feel lucky to be with so many excellent researchers in Michaelis’s group “AGMI”. Thanks are due to all colleagues of my institute, who were always quite helpful during my stay. I am expressing my sincere gratitude to Moftah Elzobi, Jörg Appenrodt, Saira Pathan and Omer Rashid for their kind help in reviewing the text. Again many thanks to my wife for her patience with me and thanks to all my friends and colleagues here in Magdeburg, Germany, and there in Egypt.

Mahmoud Elmezain
Magdeburg, Germany
November 26, 2010

Table of Contents

Dedications	i
Abstract	ii
Zusammenfassung	iv
Acknowledgement	vi
Table of Contents	vii
List of Figures	xi
List of Tables	xviii
List of Abbreviations	xix
1 Introduction	1
1.1 Gestures and Human Computer Interaction	1
1.1.1 Problem Statement	2
1.1.2 Miscellaneous Provisions	3
1.1.3 Motivation	3
1.1.4 Applications	4
1.1.5 Contributions	6
1.2 Road Map of the Thesis	7
2 Literature Review	9
2.1 Gesture Recognition	9
2.2 Related Work	10
2.2.1 Hand Gesture Recognition	10
2.2.2 Gesture Spotting	11
2.2.3 Sign Language Recognition	13
2.3 Gesture Recognition Approaches	14
2.3.1 Neural Network-Based Approach	14
2.3.2 Template Matching-Based Approach	15
2.3.3 Hidden Markov Models-Based Approach	17
2.3.4 Conditional Random Fields-Based Approach	19

2.4	Discussion and Conclusion	20
3	Fundamental Concepts	21
3.1	Color Models	21
3.1.1	<i>RGB</i> Color Model	21
3.1.2	<i>YC_bC_r</i> Color Model	22
3.2	3D Camera Model	23
3.3	Segmentation	26
3.3.1	Skin Color Modeling Using a Unimodal Gaussian	26
3.3.2	Skin Color Modeling Using Gaussian Mixture Models	27
3.3.3	Skin Probability Image	29
3.4	Classification	30
3.4.1	Hidden Markov Models	30
3.4.1.1	Elements of HMMs	31
3.4.1.2	HMMs Basic Problems	32
3.4.1.3	Topologies of HMMs	37
3.4.2	Conditional Random Fields	39
3.4.2.1	Learning Parameter for CRFs	40
3.4.2.2	Inference CRFs	41
3.4.2.3	CRFs with Hidden Variables	42
3.5	Other Techniques	42
3.5.1	Relative Entropy	43
3.5.2	Clustering Algorithm	43
3.6	Discussion and Conclusion	45
4	Isolated Hand Gesture Recognition	47
4.1	Preprocessing	48
4.1.1	Automatic Segmentation via GMMs	49
4.1.2	Depth Map	50
4.1.3	Hand Localization	52
4.1.4	Fingertip Detection	53
4.2	Tracking	55
4.2.1	Mean-shift Analysis	55
4.2.2	Trajectory Smoothing	58
4.3	Feature Extraction	59
4.3.1	Features Analysis in Cartesian Space	59
4.3.2	Features Analysis in Polar Space	61
4.3.3	Vector Normalization and Quantization	62
4.4	Classification	64
4.4.1	Classification Using HMMs	64
4.4.1.1	Model Size	65
4.4.1.2	Initializing a Left-Right Banded Model	66
4.4.1.3	Termination of HMMs Training	67

4.4.2	Classification Using CRFs	68
4.4.2.1	Data Format of CRFs	68
4.4.2.2	Matching CRFs Model	69
4.5	Computational Complexity	70
4.6	Discussion and Conclusion	72
5	Isolated Gesture Recognition Test	73
5.1	Data Set	73
5.2	Experimental Discussion	73
5.3	Experimental Results and Analysis	74
5.3.1	HMMs	75
5.3.1.1	Feature Extraction Analysis	75
5.3.1.2	Analysis Results of HMMs Topologies	78
5.3.2	CRFs, HCRFs and LDCRFs	80
5.3.3	Generative Model versus Discriminative Models	83
5.4	Discussion and Conclusion	85
6	Gesture Spotting and Recognition	87
6.1	Spotting with HMMs	88
6.1.1	Gesture Model	88
6.1.2	Non-gesture Model	89
6.1.3	Model Reduction	91
6.1.4	Gesture Spotting Network	92
6.1.5	Spotting and Recognition	93
6.2	Spotting with CRFs	97
6.2.1	Gestures and Non-gesture Model	97
6.2.2	N-CRFs Model Parameters	98
6.2.3	Forward Gesture Spotting and Recognition	99
6.3	Computational Complexity	101
6.4	Experimental Results and Analysis	102
6.4.1	Key Gesture Spotting with HMMs	102
6.4.2	Key Gesture Spotting with CRFs	107
6.4.3	Gesture Spotting with HMMs versus CRFs	113
6.5	Discussion and Conclusion	116
7	Conclusions and Future Work	119
7.1	Thesis Summary	119
7.2	Future Work	122
	Appendices:	124
A	Data Processing	125
A.1	skin and non-skin database	125
A.2	Cluster Hand Trajectory	127

A.3 Mean-shift Analysis	130
B Classification Results	131
B.1 Isolated gestures	131
B.2 Gesture spotting	135
Bibliography	139
Curriculum Vitae	151
Related Publications	153

List of Figures

1.1	Gesture spotting structure where the yellow color represents non-gesture patterns and the red color represents meaningful gesture.	3
2.1	The above samples represent the posture for alphabets from A to E, and the down samples refer to the gestures [1,2].	10
2.2	Output of CDP matching algorithm. CDP computes the optimal path and the minimum cumulative distance between the gesture models and the input sequence to detect a candidate gesture.	16
2.3	Gesture trajectory and spotting with three main phases.	18
3.1	(a) <i>RGB</i> color model. (b) <i>YC_bC_r</i> color model.	22
3.2	(a) Bumblebee stereo vision camera where its size is approximately 160 × 40 × 50 mm and consists of two Sony progressive scan CDDs color sensors with 6mm focal length. (b)The geometry of stereo camera with normal optical axes.	24
3.3	(a) Left image of video stream (b) Right image of video stream (c) Depth value of the left and right images.	25
3.4	Trellis diagram for the forward algorithm.	33
3.5	Trellis diagram for the forward algorithm, where $\delta_t(j)$ is the highest probability of landing in state j at time t after seeing the observation up to time t	34
3.6	Trellis diagram for Baum-Welch learning process. (a) The probability of traversing an arc from state i at time t to state j at time $t + 1$. (b) The probability of state i at time t	36
3.7	Ergodic model with four states.	37
3.8	Left-Right model with four states.	38
3.9	Left-Right Banded model with four states.	39
3.10	Graphical structure of a chain-structured CRFs for sequences. The variables corresponding to unshaded nodes are not generated by the model.	40
3.11	Different type of CRFs with hidden states.	42
3.12	Demonstration of k -means clustering algorithm [3].	44
4.1	Systematic concept of the isolated hand gesture recognition system. .	48

4.2	(a) Original 2D image. (b) Normalized 2D depth image. (c) Normalized 3D depth. (d) The top image represents skin pixel detection with depth value up to 10 m. In addition, the skin pixel detection without noise is represented in the bottom image (the depth value ranges from 30 cm to 200 cm). Yellow color shows skin pixels detection. F refers to the face, HL and HR represent the left and right hands respectively.	51
4.3	Skin color segmentation and hand localization. (a) Source image. (b) Labeled skin detection. (c) Hand localization with a boundary area, bounding box and centroid point.	52
4.4	Solving overlapping problem between hand and face using depth map. (a) 2D image in which the face and the left hand are occluded. (b) 2D image with labeled hands and face without occlusion.	53
4.5	Peak and valley detection. In the above graph, maximum local extreme value selects contour points SCP_1 and SCP_2 from the two clusters C_1 and C_2 . The down graph shows that the normalized values greater than 0.5 are detected as fingertip and signed by red point.	54
4.6	Fingertip detection is marked by red point for the left hand and the centroid point is marked by white point.	55
4.7	Epanechnikov kernel and histogram for the left hand which is depicted in Fig. 4.3. (a) Epanechnikov kernel for the hand target. (b) Projection of 2D weighted histogram of left hand target by using Epanechnikov kernel for (C_b, C_r) components with 16×16 bins.	56
4.8	Hand gesture path for alphabet ‘N’ using the centroid point and number ‘8’ using fingertip detection.	58
4.9	(a) Smoothing result for gesture path ‘W’, where the above curve refers to original trajectory and the down curve represents a smoothed trajectory. (b) Hand gesture path shapes for alphabets (A-Z) and numbers (0-9). Green points denote the start points of gesture path explaining the trend.	59
4.10	(a) Orientation according to the centroid of gesture path. (b) The directional codewords from 1 to 18 in case of dividing the orientation by 20°	60
4.11	Differences in velocity of gesture ‘A’ and gesture ‘K’.	61
4.12	Transformation of gesture path ‘R’ from Cartesian to Polar coordinate spaces. (a) $x-y$ space of gesture ‘R’. (b) $\rho_c-\varphi_c$ space of gesture ‘R’. (c) $\rho_{sc}-\varphi_{sc}$ space of gesture ‘R’.	62
4.13	Simplified structure shows the main processes for feature extraction stage of isolated gesture recognition system.	63
4.14	Block diagram of an isolated gestures by using HMMs (Viterbi) recognizer.	65

4.15	Straight-line segment for HMMs topologies (a) Gesture number from hand motion trajectory (b) Line segment of gesture number (c) LRB model with line segmented codewords.	66
4.16	Block diagram of an isolated gesture using CRFs recognizer.	69
5.1	IESK lab.	74
5.2	The number of feature codes represents either the number of clusters in case of combined features or the number of normalized codewords in case of separated features. (a) The recognition of locations and velocity features according to different number of codewords (10, 15, 20, 25, 30). (b) Results for three different orientations with varying feature codewords number (9, 12, 18, 36). (c) Recognition rate of different combined features in Cartesian system with different codewords number ranging from 28 to 37.	76
5.3	(c) Recognition rate according to combined features in Polar system with different feature codewords number ranging from 28 to 37. (b) The highest priority at $t = 21$ is gesture number '2' and at $t = 47$ the final result is gesture number '3'.	78
5.4	Isolated gesture recognition results for HMMs topologies with number of states ranging from 3 to 10.	79
5.5	Recognition accuracy with different window sizes (0-7) for CRFs, HCRFs and LDCRFs on training and testing data.	80
5.6	Temporal evolution of the seven higher probabilities of the gestures 'B', 'F', 'K', 'M', 'P', 'R' and 'T' using CRFs. In the image sequences, the high priority is alphabet 'F' at $t = 28$, at $t = 45$ the high priority is alphabet 'P' and at $t = 70$ the result is 'R'. The hand motion trajectory is generated by connecting the centroid point of hand region.	81
5.7	Temporal evolution of the seven higher probabilities of the gestures 'B', 'R', 'Z', '2', '3', '7' and '8' using HCRFs. In the image sequences, the high priority is number '2' at $t = 24$, at $t = 40$ the high priority is number '8' and at $t = 53$ the result is '8'. The hand motion trajectory is generated by connecting the fingertip points of the region of interest.	82
5.8	Results of gestures recognition using CRFs, HCRFs, LDCRFs versus HMMs at window size = 4.	84
6.1	Concept of the hand gesture spotting and recognition system.	88
6.2	Road map of gesture spotting and recognition using HMMs.	88
6.3	The hand gesture paths and straight-line segmentation. (a) The gesture paths from hand motion trajectory for numbers (0-9) with its segmented parts. (b) The LRB topology with segmented line for a gesture path '4'.	89

6.4	(a) Ergodic topology (b) Simplified ergodic with two dummy states and fewer transitions.	90
6.5	The general non-gesture model where the dotted arrows represent null transitions, $G_{i,j}$ refers to the state j for gesture number i , ST and ET are the two dummy states for starting and ending, receptively.	91
6.6	The gesture spotting network which contains ten number gesture models from 0 to 9 and are designed by using LRB model with varying states from 3 to 5 and the Non-gesture model.	93
6.7	Simplified structure showing the main modules for hand gesture spotting via HMMs, where the start and end points are based on differential probability value.	94
6.8	Block diagram shows the work of sliding window. The Viterbi algorithm recognizes the segmented parts after detecting the start point.	96
6.9	Road map of gesture spotting and recognition using CRFs.	97
6.10	Simplified structure showing the main modules for hand gesture spotting via CRFs.	100
6.11	Temporal evolution of four higher probabilities of the gestures ‘3’, ‘9’, ‘Non-gesture’ before and after state reduction. The probability of Non-gesture model before and after state reduction is the same. In the image sequences, the high priority is gesture ‘3’ and the second priority refers to Non-gesture ‘N’ at $t = 24$. The final result is gesture number ‘3’ at $t = 42$	103
6.12	(a) Spotting accuracy using HMMs relative to sliding window size ranging from 1 to 8. (b) Insertion, deletion and substitution errors relative to sliding window size.	105
6.13	Image sequences contain one meaningful gesture ‘6’, where the start point at frame 15 and the end point at frame 50. ‘N’ refers to Non-gesture.	106
6.14	(a) Spotting accuracy using CRFs relative to sliding window sizes (1-8). (b) Insertion, deletion and substitution errors relative to sliding window size.	109
6.15	Temporal evolution of gesture ‘3’ and non-gesture probabilities.	109
6.16	Temporal evolution of the probabilities of the gesture numbers (0-9) and non-gesture label ‘N’. The image sequences contain one key gestures ‘3’, where the start point is at frame 19 and the end point is at frame 51. In the first 18 frames, the probability of non-gesture label is assigned the greatest value, which means that the start point of the key gesture is not detected. At frame 19, the start point is detected since the higher priority is assigned to gesture labels than the non-gesture label. At frame 51, the higher priority is non-gesture label which means that the end point of key gesture ‘3’ is detected.	110

6.17	Temporal evolution of the probabilities of the gesture numbers (0-9) and non-gesture label 'N'. The image sequences contain two key gestures '3', '2', where the end point of meaningful gesture '3' at frame 66 and the start point of meaningful gesture '2' at frame 85. Between frame 67 and frame 85, the higher priority is assigned to non-gesture label which means that the start point of the second key gesture is not detected. At frame 86, a new key gesture is started where the probability value of non-gesture label is not the highest value as compared to the other gesture labels.	111
6.18	Temporal evolution of the probabilities of the gesture numbers '2', '3', '6' and non-gesture label 'N'. The image sequences contain three key gestures '3', '2', '6'. The end point of meaningful gesture '3' is at frame 37. Between frame 37 and frame 50, the higher priority is assigned to non-gesture label which means that the start point of the second key gesture is not detected. At frame 51, a new key gesture has started where the probability value of non-gesture label is not the highest value as compared to the other gesture labels. The end point of meaningful gesture '2' is at frame 75. Between frame 75 and frame 91, the higher priority is assigned to non-gesture label. The start point of meaningful gesture '6' is at frame 92. The final result of the continuous gesture path is '326'.	112
6.19	A comparison result between HMMs and CRFs. (a) Error types (Insertion: I, Deletion: D, substitution: S) of CRFs and HMMs. (b) The recognition and the reliability of HMMs and CRFs where the reliability of system considers the insertion error in calculation.	114
6.20	Average segmentation time of forward and backward spotting method.	116
A.1	Cropped images for skin and non-skin pixels that were collected from the World Wide Web. (a) Database of skin pixels for different races. (b) Database of non-skin pixels for different background.	125
A.2	Distribution values of skin and non-skin pixels projected onto the (C_b, C_r) plane for training data. (a) Distribution values of skin pixels for training data where the skin color is localized to a small region in the (C_b, C_r) chrominance space. (b) Location of the mean points according to three components of Gaussian Mixture Models for skin database. (c) Non-skin pixels distribution for training data.	126
A.3	Cluster trajectories in Cartesian system for gesture numbers according to $(Lc, Lsc, \theta_1, \theta_2, \theta_3, V)$ features. The middle and bottom graphs are the same of the top graph after eliminating the different cluster trajectories. Here, gesture paths '0' and '6' have the same cluster indices until frame 33.	127

A.4	Cluster trajectories for gesture ‘3’ and gesture ‘5’ according to features $(Lc, Lsc, \theta_1, \theta_2, \theta_3, V)$, (ρ_c, φ_c) and $(\theta_1, \theta_2, \theta_3)$, respectively. The cluster trajectories which are depicted in the middle and bottom graphs are varying than the top graph, notably in the later parts of gesture paths ‘3’ and ‘5’.	128
A.5	Cluster trajectories for the gesture path ‘3’ with respect to different five video samples. It is noted that the same gesture have similar cluster indices but with slight variations in their cluster trajectories (i.e. spatio-temporal variabilities).	129
A.6	Tracking result where at frame 109, both hands are correctly determined notably in case of overlapping and partial occlusion. In top figure, the number of mean-shift iteration is 1.61 per frame for both left and right hands, which in turn makes the system capable for real-time implementation.	130
B.1	Hand gesture paths for gesture numbers from 0 to 9 with segmented parts.	131
B.2	Hand gesture paths for alphabets from A to M with segmented parts.	132
B.3	Hand gesture paths for alphabets from N to Z with segmented parts.	133
B.4	Temporal evolution of the seven higher probabilities of the gestures ‘C’, ‘G’, ‘S’, ‘0’, ‘4’, ‘5’ and ‘6’ using LDCRFs. In the image sequences, the highest priority is gesture number ‘6’ at frame 21 as well as in frame 31, and at frame 36 the result is gesture number ‘6’.	134
B.5	Temporal evolution of the probabilities of the gestures number (0-9) and non-gesture label ‘N’. The image sequences contain one meaningful gestures ‘6’. At frame 15, the start point is detected since the highest priority is assigned to gesture labels than the non-gesture label. At frame 50, the highest priority is assigned to non-gesture label which means that the end point of meaningful gesture ‘6’ is detected.	135
B.6	Temporal evolution of the probabilities of the gestures number (0-9) and non-gesture label ‘N’. The image sequences contain two key gestures ‘6’ ‘2’, where the end point of gesture ‘6’ is at frame 56 and the start point of gesture ‘2’ is at frame 76. In the first 55 frames, the probability of non-gesture label is not the maximum value, which means that the end point of the key gesture is not detected. At frame 56, the first key gesture ‘6’ ends where the non-gesture label has a high probability than other gesture labels. Between frame 56 and frame 75, the highest priority is assigned to non-gesture label, which means that the start point of the second key gesture is not detected. At frame 76, a new key gesture is started, where the probability value of non-gesture label is not the highest value as compared to the other gesture labels.	136

- B.7 Temporal evolution of the probabilities of the gestures number ‘4’, ‘5’, ‘8’ and non-gesture label ‘N’. The image sequences contain three key gestures ‘5’, ‘8’, ‘4’. The end point of gesture ‘5’ is at frame 41. Between frame 42 and frame 56, the highest priority is assigned to non-gesture label, which means that the start point of the second key gesture is not detected. At frame 57, a new key gesture is started where the probability value of non-gesture label is not the highest value as compared to the other gesture labels. The end point of gesture ‘8’ is at frame 93. Between frame 94 and frame 102, the highest priority is assign to non-gesture label. The start point of gesture path ‘4’ is at frame 103. The final result of the continuous gesture path is ‘584’. . . 137
- B.8 Temporal evolution of the probabilities of the gestures number ‘5’, ‘6’, ‘7’ and non-gesture label ‘N’. The image sequences contain three key gestures ‘7’, ‘6’, ‘5’. The end point of gesture ‘7’ is at frame 21. Between frame 22 and frame 34, the highest priority is assigned to non-gesture label, which means that the start point of the second key gesture is not detected. At frame 35, a new key gesture is started where the probability value of non-gesture label is not the highest value as compared to the other gesture labels. The end point of gesture ‘6’ is at frame 57. Between frame 58 and frame 73, the highest priority is assign to non-gesture label. The start point of gesture path ‘5’ is at frame 74. The final result of the continuous gesture path is ‘765’. . . 138

List of Tables

4.1	Gaussian mixture model for skin color database which contains the mean vector, covariance matrix and mixture weight for $K = 3$ clusters.	49
4.2	Unimodel Gaussian for non skin color.	50
5.1	Results of isolated gestures according to different features extraction in Cartesian and Polar systems with optimal feature code number. . .	77
5.2	Results of gestures recognition at $W = 0$	84
6.1	Isolated gesture recognition and key spotting gesture results for gesture numbers from '0' to '9' using HMMs at sliding window equal to 5. . .	104
6.2	Results of isolated gestures recognition and key gestures spotting with different size of sliding window (Sw) ranging from 1 to 8 via HMMs. .	106
6.3	Results of isolated gestures recognition and key spotting gestures with different size of sliding window (Sw) ranging from 1 to 8 using CRFs.	108
6.4	Results of spotting key gestures using HMMs versus CRFs.	115

List of Abbreviations

Nomenclature	Description
HCI	Human Computer Interaction
HMMs	Hidden Markov Models
cHMMs	Coupled Hidden Markov Models
CRFs	Conditional Random Fields
HCRFs	Hidden Conditional Random Fields
LDCRFs	Latent-Dynamic Conditional Random Fields
MEMMs	Maximum Entropy Markov Models
DP	Differential Probability
CDP	Continuous Dynamic Programming
DTW	Dynamic Time Warping
ASL	American Sign Language
GSL	Greek Sign Language
JSL	Japanese Sign Language
FSL	French Sign Language
NNs	Neural Networks
GIVEN	Gesture driven Interface in Virtual Environments
HAS	Hierarchical Activity Segmentation
<i>RGB</i>	Red Green Blue
<i>nRGB</i>	Normalized <i>RGB</i>
<i>HSI</i>	Hue Saturation Intensity
<i>HSV</i>	Hue Saturation Value
GMMs	Gaussian Mixture Models
SOM	Self Organizing Maps
EM	Expectation Maximization
ML	Maximum Likelihood

Nomenclature	Description
BW	Baum-Welch
LR	Left-right
LRB	Left-right Banded
F	Face
HL	Hand Left
HR	Hand Right
pdf's	probability density function
CSV	Comma Separated Values
BFGS	Broyden-Fletcher-Goldfarb-Shanno
FPS	Frame Per Second
<i>Sw</i>	Sliding window
I	Insertion
S	Substitution
D	Deletion
MAD	Mean Absolute Difference
NCC	Normalized Cross Correlation
ROI	Region of Interest
SCP	Selected Contour Point
CP	Center Point

Chapter 1

Introduction

1.1 Gestures and Human Computer Interaction

The process of communication is to transfer information from one entity to another. Naturally, hand gestures are powerful human-to-human communication channel, which forms a major part of information transfer in our everyday life. There are many ways to perform and interpret a human action using either hands and/or arms. A gesture is a spatio-temporal pattern which may be static, dynamic or both¹. The performance of computers could be greatly enhanced if they were able to recognize gestures and their interaction with humans to be more “human-like”. Keyboards, mouses and joysticks are the most commonly used devices to deal with computer until now. The interaction with the computer is emerged as a new research field called Human Computer Interaction (HCI). The main theme of HCI is to propose new methodologies and techniques to improve the interaction between humans and computers. Researchers have exploited and combined different interfaces between humans and computers which include both software and hardware components. The initial attempts focused on the interpretation of languages allow the understanding of human linguistics. Moreover, many hand recognition systems have been proposed, which in turn play an important role in this area.

Recently, the focus of HCI is shifted to visual interaction with computers through virtual interfaces, haptic interfaces and virtual reality [4]. The main goal is the interaction of humans virtually by the analysis of hand or body movements in 3D space which is not possible with traditional 2D devices. The technological development is a major achievement for HCI because it provides all the means of support and comfort to interact with machines. Despite of the restricted area of computer vision with HCI, it is an attractive area of research to invent new methodology of interaction between humans and machines. Until now, the natural interaction is a challenge for the research and is yet to be addressed. So, there is an intensive research in the field of

¹Static morphs of the hands are called postures and hand motions are called gestures.

vision based gesture recognition. Therefore, many approaches have been proposed to solve the research challenges in various commercial applications (e.g. gesture control mobile interface and remote control). Additionally, the rapid technological development in hardware and software (i.e. high processing, high capacity and low cost for programs development) enhances the research in this field dramatically.

Hand gestures are easy to use and more convenient for humans to interact with computers. For example, sign languages are considered as one of the main applications areas which have been used among the deaf people (i.e. speech-disabled people) [5]. In addition, the people with the ability to speak also use gestures in order to communicate with each other. There are many successful applications of hand gesture recognition like human-robot interaction [6], television control and computer game [7], video annotation and indexing [8], and video surveillance [9]. In this thesis, we focus on the problems of extracting meaningful motion patterns from input video stream.

1.1.1 Problem Statement

The task of extracting meaningful patterns from input signals is called pattern spotting [10, 11]. In gesture spotting, an instance of pattern spotting is required to locate the start point and the end point of a gesture (Fig. 1.1). The gesture spotting has two major challenges that arise in hand gesture recognition: segmentation [5, 12] and spatio-temporal variabilities² [13, 14]. The segmentation problem is about determining the start and the end point of the gesture in a continuous hand trajectory. As the user switches from one gesture to another, his hand makes an intermediate move linking the two consecutive gestures. A gesture recognizer may attempt to recognize this inevitable intermediate motion as a meaningful one. The other difficulty of gesture spotting is caused because the same gesture varies dynamically in shape, trajectory and duration even for the same person. Therefore, the recognition step should consider both the spatial and temporal variabilities simultaneously. A robust recognition phase extracts the gesture segments from the input signal and match them with the reference patterns regardless of the spatio-temporal variabilities. Additionally, previous approaches [15, 16, 17] mostly use the backward spotting technique to first detect the end point of gesture. Secondly, it tracks back to discover the start point of the meaningful gesture through its optimal path and the segmented gesture is sent to the classification phase for the recognition. In these approaches, there is an inevitable time delay between the meaningful gesture spotting and recognition. This time delay is unacceptable for online applications. Above all, few researchers have addressed the problem of non-sign patterns, which include out-of-vocabulary signs and other movements that do not correspond to signs. Hence, there is a difficulty in building a model for non-sign patterns where the set of them is unknown for great diversity.

²Spatio means determining *where* the hand gesture is located at each frame. Temporal means determining *when* the gesture starts and ends.

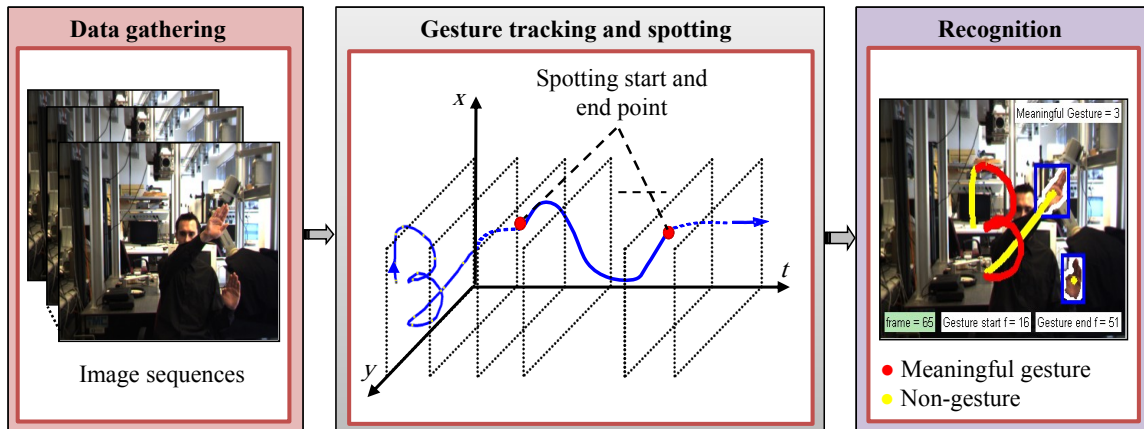


Figure 1.1: Gesture spotting structure where the yellow color represents non-gesture patterns and the red color represents meaningful gesture.

1.1.2 Miscellaneous Provisions

Every field has some problems and challenges that needs to be addressed. Similarly, in the research area of hand gesture spotting and recognition, there are many issues for data gathering, segmentation, gesture tracking and feature selection which still need to be improved. These include:

- If the hand motion is fast in front of a normal camera system, it leads to many problems, foremost is the segmentation problem and false detection of the gesture (i.e. reconstruction problem of the movement).
- For each type of gesture, there is a quite bit of variability (even for the same person) in terms of the pose of hand, the speed and duration of conducted gesture and lapse of trajectory.
- How to select the optimal features of hand gesture model taking in account the challenges that are faced by the generation parameters as rotation and scaling.
- In the case of acquisition failure of depth map sequences when projecting a 3D scene to 2D plane, the reconstruction of the hand trajectory is almost impossible dues to the existence of serious shortcomings in the segmentation process.
- How to segment the meaningful gestures (gesture spotting) that are introduced into the system for the same continuous hand movements.

1.1.3 Motivation

The latest advancements in computer vision and computer hardware technologies make the research of real-time hand tracking and gesture recognition promising. However, many current approaches still suffer from the limitation of accuracy, robustness

and speed. This makes the gesture interaction indirect and unnatural. The objective of this work is to build a real-time capable system for hand tracking, gesture spotting and recognition. To achieve this goal, an application of image-based interaction with alphabets (A-Z) and numbers (0-9) is considered to be the domain of the system. Our system is built in a way that it focuses mainly on hand gesture spotting and recognition without using colored gloves or markers. Moreover, the system uses a stereo camera system for the image acquisition. The investigation of existing research led us to make the following assertions:

- **Real-time performance:** For the real-time performance, the system is able to analyze the image sequences at any frame rate with minimum process speed to give the user instant feedback of the recognized gesture. In addition, the system should be robust against the issue of time delay at any phase throughout the process of gesture spotting and recognition.
- **Accuracy:** A hand gesture spotting and recognition system should be able to tolerate some mistakes and errors such as spatio-temporal variabilities. However, it still needs to be accurate enough in order to be viable. For instance, the system should achieve a higher detection rate while maintaining a low false positive rate for each gesture. Moreover, the system should spot and recognize different gestures without confusion among them.
- **Robustness:** The system should track the hand when applied on several video sequences containing confusing situations such as partial occlusion and overlapping. Additionally, the hand gesture should be robustly recognized under different illumination conditions and cluttered backgrounds.
- **Scalability:** The system is able to deal with a small or a large gesture vocabulary by adding specific requirements (for instance, adding short gesture detector in case of gesture spotting). Thus, it is practical and efficient when applied to different situations like spatio-temporal variabilities.
- **User-independence:** The system should spot and recognize hand gestures with different shapes, skin colors, trajectories and durations. The system should also have the ability to deal with the movement of the hand signs for different users rather than a specific user.

1.1.4 Applications

In the field of HCI, hand gesture recognition domain is a big challenges for researchers. Moreover, the gesture spotting and interpretation are essential to make the human-machine communication close to human-human interaction. Application areas for gesture interactions include HCI, computer games and intention analysis. Furthermore, an important area for gesture interaction lies in the recognition of sign language.

In addition, gesture recognition by computer offers new applications in industry (for instance, steering and control of robots) and in security (e.g. event recognition). In the following, some of the most active application areas of gesture recognition are described.

- **Human Computer Interaction:** HCI is a successful application to recognize the meaningful gestures from continuous video. The main goal of HCI is to make the interaction between human and computer running normally. Specifically, HCI is usually designed to interact with the practical applications of real-world problems (e.g. computer access for deaf peoples and control virtual environments etc.).
- **Human Robot Interaction:** Robots usually reach or manipulate objects using their mechanical parts (e.g. equivalent to hand and arm). Gestures can be used to control such movements. Also, to move in the physical world, robots need guidance, therefore gestures can be easily used for such purposes.
- **Television Control and Computer Games:** The TV control is one of the important applications for hand gestures. Hand gestures provide the user with an appropriate speed for the various operations on TV (e.g. increase/decrease the volume and TV on/off etc.). Another application is to play games, where the hand gestures are used as an interacting modem in order to control the games easily.
- **Sign Language Recognition:** Sign language recognition is considered as one of the intuitive applications for hand gestures. There are many useful applications in our daily lives, which are based on the analysis of sign language. Some of these applications include: sign-to-text, translation from one language to other languages and vice versa.
- **Gesture-to-Speech:** By the gesture-to-speech application, the hand gestures are analyzed and translated into speech. This type of application is important for the people who are not fluent in sign language expression.
- **Intention Analysis:** The intention analysis deals with the recognition of words and alphabets before they are completed. Furthermore, the intention system inform the user about the successful goal before the gesture ends (i.e. predicting of the event before it happens).
- **Virtual Reality:** Virtual reality interactions are applied to computer-simulated environments which are similar to the real-world interaction (e.g. simulation of the combat training). The users can interact with the virtual environment using the interpretation of hand gestures as an input device for 3D display interactions.

1.1.5 Contributions

To face the mentioned challenges, a forward gesture spotting method is proposed, which simultaneously handles the hand gesture spotting and recognition in stereo color image sequences without time delay. To spot meaningful gestures accurately, stochastic methods for designing a non-gesture model with Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) are proposed with no training data. The non-gesture model provides the confidence measure and is used as an adaptive threshold to find the start and the end points of meaningful gestures which are embedded in the input video stream. To demonstrate the coaction of the suggested components and the effectiveness of our gesture spotting and recognition system, an application of gesture-based interaction with alphabets (A-Z) and numbers (0-9) is implemented. The major contributions of this thesis are presented as follow:

- **Depth map:** One of the main contributions of this work is to exploit depth image sequences. The main motivation behind the use of depth information is to identify the Region of Interest (ROI) without processing the whole image, which consequently reduces the cost of ROI searching and increases the processing speed. Furthermore, the depth information is used to resolve complex backgrounds (i.e. neutralize complex background) completely, as well as illumination variation and it also increases the accuracy of objects segmentation. In the case of overlapping (i.e. ambiguities) between the hands and face, the depth information is used to identify the objects under occlusion.
- **Hand tracking and feature extraction:** A robust method for hand tracking in a complex environment using Mean-shift algorithm in conjunction with depth map is proposed. This scheme correctly extracts a set of hand postures to track the hand motion and achieves accurate and robust real-time hand tracking. Features like location, orientation and velocity (which are obtained from spatio-temporal hand gesture path) with respect to Cartesian and Polar coordinate systems are combined and analyzed. This analysis determines the degree of effectiveness of these combinations on the recognition results.
- **Isolated gesture recognition:** The isolated gestures are handled according to two different classification techniques: generative model such as HMMs and discriminative models like CRFs, Hidden Conditional Random Fields (HCRFs) and Latent-Dynamic Conditional Random Fields (LDCRFs). Additionally, different HMMs topologies are analyzed and studied in terms of their impact on isolated gesture recognition. This research is focused on the decision of HMMs topology and classification techniques for the optimal results.
- **Gesture spotting and recognition:** A stochastic method with no training data for designing a non-gesture model by HMMs and CRFs is proposed to

spot meaningful gestures accurately. The non-gesture model provides confidence measure and is used as an adaptive threshold. The main motivation of using this adaptive threshold is to find the start and the end points of meaningful gestures from hand continuous motion.

- **Improving the performance of gesture spotting and recognition:** The non-gesture model with HMMs is modified by using relative entropy function to cure the problem of increasing number of states. The main objective is to save time and space, and to increase the spotting speed. Another modification for non-gesture model with CRFs is to add a short gesture detector, which increases the weights of self-transition feature functions for short gestures to further improve the accuracy of gesture spotting. The effectiveness of improved non-gesture model yields reasonable recognition rates. In addition, it is robust against errors³ which are caused by spatio-temporal variabilities.
- **Forward spotting:** The drawback of the backward spotting technique is the time delay between gesture segmentation and recognition, in which it has to spend additional time for backtracking to find the gesture start point. In order to solve this problem, a forward spotting method in conjunction with the sliding window technique is proposed to handle hand gesture segmentation and recognition simultaneously. The main objective is to resolve the following issues; avoiding the time delay between meaningful gesture spotting and recognition, achieving accurate, robust as well as making the system capable of working for on-line applications.

1.2 Road Map of the Thesis

The thesis is structured in seven chapters as follows:

- Chapter 1 presents the relationship between gestures and HCI. The major challenges of gesture spotting problem are described. In addition, the motivation and the contribution of the work are also given.
- Chapter 2 surveys the literature of hand gesture. The chapter starts with an overview of the research highlights and the challenges which are present in the research field from the aspect of three main points: 1) Hand gesture recognition 2) Gesture spotting 3) Sign language recognition. After that, the major approaches which include Neural Networks, template matching, HMMs and CRFs are summarized. These approaches give more attentions to analyze and extract patterns with spatial and temporal variabilities. Moreover, this chapter is important in the context of understanding the motivation of doing the research and enables to investigate and compare the novel techniques.

³More details about these errors can be found in Section 6.4.1.

- Chapter 3 gives an insight into the fundamental concepts which build the bases for understanding this thesis. Firstly, the color models like RGB , YC_bC_r are discussed with some details. Secondly, normal Gaussian distribution and Gaussian mixture models are presented for the segmentation of hands and face. After that, the classification approaches (i.e. HMMs and CRFs) are explained. Lastly, relative entropy and k -means algorithm are summarized which are used to improve the hand gesture recognition.
- Chapter 4 describes the proposed isolated gesture recognition system in four main phases: preprocessing, tracking, feature extraction and classification. The object segmentation and tracking under occlusion are exploited with 3D depth map. To motivate the extracted feature of gestures in this chapter, dynamic features with respect to Cartesian and Polar coordinate systems are presented. After that, major classification techniques based on HMMs, CRFs, HCRFs and LDCRFs are discussed.
- Chapter 5 demonstrates the effectiveness of the isolated gesture recognition system for HCI. This chapter examines the capabilities of combined features of location, orientation and velocity for gesture recognition with respect to Cartesian and Polar coordinate systems. In addition, the effective of these features are presented which yields reasonable recognition rates. The experiments are carried out on isolated gestures (alphabets and numbers) according to two different classification techniques: a generative model such as HMMs and discriminative models like CRFs, HCRFs and LDCRFs. Comparison of results between generative and discriminative models are also provided.
- Chapter 6 describes the spotting system which is used to extract meaningful gestures from the input video stream. The set of spotting rules which are used in the system are derived according to HMMs and CRFs. This chapter presents the modelling of gesture patterns discriminately and non-gesture model effectively with no training data for non-gesture patterns. To motivate the gesture spotting problem and to solve the issues of time delay between the gesture segmentation and recognition, a forward spotting scheme is presented which uses a stochastic method for designing a non-gesture model with HMMs or CRFs models. Moreover, the concept of relative entropy with HMMs and short gesture detector with CRFs are introduced in order to improve the gesture spotting system's accuracy. At the end, the quantitative experiments conducted with the proposed system and the performance measures used for their evaluation are discussed.
- Chapter 7 concludes the thesis by summarizing the contributions of this work as well as the possible improvements for future work.

Chapter 2

Literature Review

2.1 Gesture Recognition

In recent years, the hand gesture recognition has become a major research challenge due to its large use in HCI, image/video coding, and content-based image/video retrieval. For example, a successful hand gesture recognition system provides valuable insight into how one might approach other similar pattern recognition problems such as facial expression interpretation, lip reading and human action identification. Generally speaking, gestures are predefined paths that have a symbolic meaning. They can be made in either 2D or 3D space using a suitable input device. 2D gestures are usually drawn with a mouse or stylus on a tablet. For the purpose of this proposal, when we consider gestures in 3D, they tend to be hand gestures made with a suitable hand tracking device like the glove device or the camera. The gestures can be classified into two classes according to the inclusion of the hand motions. The first one is called posture in which static hand positions stay in the same space whereas movement of dynamic hands and fingers are referred as gestures as shown in Fig. 2.1. Human-human communication acts a basis to develop human-computer communication which is a considerable approach for more natural communication with the computer. Communication between humans is often inaccurate which is usually expressed by using hand gestures [15]. Entrance gesture sometimes be appropriate for people who are unable to use a keyboard because they fear of using the keyboard and may prefer having the system which responds to gestures guide. Vision based analysis of hand gestures is the most natural way of constructing a human-computer gestural interface. One or more cameras are used to capture hand motion for vision based gesture recognition. Many vision techniques are applied to real-time video stream of user gestures with unadorned hand [17,18]. That is, the user can move his hand without any equipment, and camera captures video frames of a user, and then vision techniques are used to extract hand from the video frames. With cameras, the user can conduct raw hand gestures as in human-human communication. However, vision based systems require much more computing power for real-time applications because

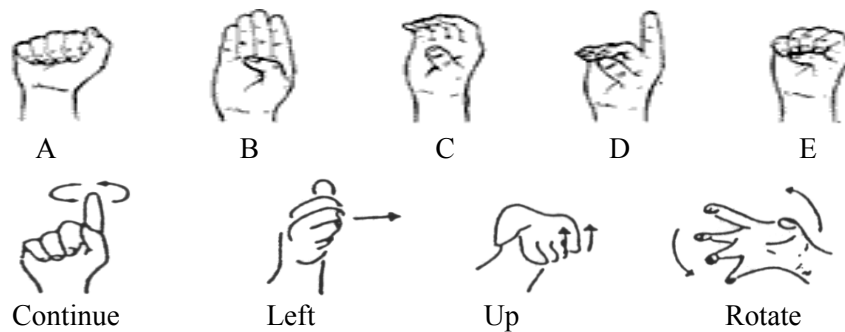


Figure 2.1: The above samples represent the posture for alphabets from A to E, and the down samples refer to the gestures [1, 2].

vision based techniques used for hand tracking are time consuming processes. One of the main advantages for studying the hand gestures is that it is a step towards the recognition of sign language. In order to realistically expect hand gestures to be used for HCI, the gesture recognition module not only accurate but also as part of a large system, it must be efficient since only a small portion of system resources is devoted to the module. Hence, many of the design approaches like HMMs and CRFs are constructed in favor of faster computation even if there is negligible degradation of recognition performance.

2.2 Related Work

Pattern spotting is the process of segmenting meaningful patterns from input streams and identifying them according to the classification technique. Spotting of sign language is considered an instance of pattern spotting. There are many different techniques and applications for pattern spotting which have been taken in account in this work. This section briefly reviews the representative pattern spotting techniques in addition to the related works from many different areas: hand gesture recognition mainly from the area of computer science and HCI, and gesture spotting mainly from the area of computer vision, artificial intelligence and speech recognition.

2.2.1 Hand Gesture Recognition

Hand gestures represent sequential data which can vary in both trajectories and durations. A common gesture recognition technique is used to deal with gestures as an output of observable task so that it holds the property of Markov. In Markov models, the conditional probability density function of current state is based only on recent states. HMMs are considered as one of this architecture and is employed as a

probabilistic network with hidden and emission states. HMMs are the most common approach used for gesture classification to score remarkable success in modeling spatio-temporal time series [15, 19]. In [20], an HMM was employed to recognize the tracked gesture for control desktop applications like games, painting programs and browsers. In [21], an application system by recognition of HMMs is integrated to health center in which the patients used colored gloves to express their needs to the centralized system. Instead of using colored gloves, Vogler and Metaxas [22, 23] use 3D object shape and motion extracted features with computer vision methods as well as a magnetic tracker fixed with the signer's wrists. They introduce a parallel algorithm using HMMs in order to model and recognize gestures from continuous input stream. Shape, location and trajectory of left hand, in addition to location and trajectory of right hand are implemented using separate channels of HMMs. Each channel has been learned with relevant data and combined features. Moreover, individual networks of HMMs have been constructed for each channel and a modified Viterbi algorithm was employed to search within the networks in parallel. From each network, the trajectory probabilities with the same word sequences are combined together. Tanibata *et al.* [24] proposed a similar scheme for isolated word recognition in the Japanese Sign Language (JSL). The authors apply HMMs to model the gesture data from right and left hand in a parallel mode. The information is merged by multiplying the resulting output probabilities.

In HMMs, the current observations are based only on the current state, but the current observations for the Maximum Entropy Markov Models (MEMMs) that is proposed by McCallum *et al.* depend on the previous and the current states [25]. Although MEMMs use a directed graphical model such as HMMs, it suffers from the bias problem because its states are locally normalized. CRFs are undirected graphical model and use a single exponential model for the joint probability of state sequences for a given observation sequences [26]. Let us denote the observation sequence as x and the class label or hidden state sequence as y . Then, generative models specify $p(y, x)$, the joint probability distribution over observation and class label sequences, whereas discriminative models specify $p(y|x)$, the likelihood of the label sequence conditioned on the input sequence. For sequence data, the most common generative and discriminative approaches are HMMs and CRFs, respectively. CRFs do not have the ability to learn the latent dynamics of gestures. HCRFs and LDCRFs are CRFs variant which incorporate hidden states variables to deal well with gesture substructure [27].

2.2.2 Gesture Spotting

In the gesture recognition system, one of the difficult problem is gesture spotting which means how to find the start and the end points of meaningful gestures in a continuous input stream. In general, natural input consists of gestures and non-gestures.

Non-gestures represent other movements which do not correspond to gestures such as manipulative and coarticulatory gestures. In the literature of gestures spotting, several methods were proposed for this purpose but without using the temporal segmentation [28, 29, 30, 31]. Many existing methods have been performed under the terms of codified (i.e. hands are unambiguously tracked in image sequences). Yet, this leaves quite a bit of temporal variability in hand gestures and provides a challenge for gesture spotting. A trade-off between the complexity of gesture recognition and the naturalness of performing gestures must be made. After considering the existing methods for gesture spotting, we found that these methods are classified into two approaches: the direct approach and the indirect approach. The temporal segmentation in direct approach precedes the recognition task of gestures. Direct approach is based on either low-level or mid-level motion parameters to spot gestures. Acceleration, curvature of trajectory and velocity have been employed as low-level motion parameters [32] while the activity of human body was considered as mid-level motion parameter [33]. Consequently, abrupt changes (for instance, zero-crossings) in these parameters were used as a main rule to identify meaningful gesture boundaries. The drawback of such methods is to obtain a gesture first, and then followed by specific intervals for non-gestures. As a result, these methods reflect unacceptable conditions in continuous gesturing for scientific research.

Temporal segmentation in indirect approach is interwoven with recognition task where indirect methods provide good recognition scores for the detected gesture boundaries. The work mechanism for most indirect methods [28, 29, 34] are based on the extension of dynamic programming such as Continuous Dynamic Programming (CDP) [29], Dynamic Time Warping (DTW) [35, 36], HMMs [5, 37, 38] and CRFs [16, 26, 39]. In these methods, the end point of meaningful gesture is found by comparing its likelihood score to a static or an adaptive threshold which is estimated by a non-gesture filler model as in signal processing field [15, 40]. Most existing systems are based on the use of fixed likelihood threshold to spot gestures, so that the gestures are refused when their likelihood does not exceed the allocated score to spotting threshold. Reliance on the use of a fixed threshold is considered as naive and non-practical solution to handle the likelihood variabilities computed by models. An HMM-based framework is proposed by Lee and Kim [15] which handles gesture spotting and recognition effectively using adaptive threshold to distinguish between gesture and non-gesture patterns. The non-gesture model is constructed by considering all reference states of the trained HMMs in the system (i.e. considers all reference observations probabilities, self-state transitions and ignores state transition probabilities). Furthermore, the non-gesture model provides a good confirmation for the rejection of non-gesture patterns where its likelihood is smaller than the dedicated model for a given gesture. Whereas, Yang *et al.* [16] proposed a threshold model based on CRFs, which uses an adaptive threshold to spot and recognize gestures in continuous input streams. A major limitation of such methods is that they used the

backward spotting technique to first detect the gesture end point. After that, they track back to discover the gesture start point and then the segmented gesture is sent to the recognizer for recognition. Moreover, there is a time delay between gesture spotting and recognition and this time delay is unacceptable for online applications.

2.2.3 Sign Language Recognition

Recognition of hand gesture is an active topic of research in computer vision especially for the purpose of HCI and sign language. In contrast to gestures, a typical component of spoken languages, the sign languages present the natural way for communication among deaf people. Sign languages develop, like oral languages, in a self-organized way. An example which shows that sign language appears wherever communities of deaf people exist is reported by [41]. Three problems should be solved to recognize sign language. The first challenge is the reliable tracking of the hands, followed by robust feature extraction as the second problem. Finally, the third task concerns the interpretation of the temporal feature sequence. The performance of the sign language can be divided into manual (hand orientation, location and trajectory) and non-manual (head, mouth and facial expression) parameters. Sometimes, the use of manual parameters is enough to distinguish some signs but there are ambiguities in other signs which require non-manual information to identify them.

Hienz *et al.* [42], and Bauer and Kraiss [43] proposed an HMM-based continuous sign recognition system where the signs have been divided into subunits for recognizing separately. They simplified the extracted features from image segmentation using different color gloves for hand palm and fingers. Thus, the vector sequences of extracted features reflect the manual parameters of sign. By using the same group, another system to recognize continuous signs has been constructed based on HMMs. they have used skin color detection with multiple tracking hypothesis to extract geometric features such as compactness, eccentricity and axis ratio [44, 45]. The winner hypothesis is determined at the end of the sign. However, the authors include high level knowledge of the human body and the signing process in order to compute the likelihood of all hypothesized configurations per frame.

Vassilia *et al.* [1] proposed a system to recognize both isolated and continuous Greek Sign Language (GSL) sentences for hand postures. The orientation codeword is extracted from images and is then employed in sentences for input to HMMs. Nianjun *et al.* [46] proposed a method to recognize all 26 letters from A to Z by using different HMMs topologies with different states. Nguyen *et al.* [47] proposed a real-time system to recognize 36 hand vocabularies like American Sign Language (ASL) and digits in unconstrained environments. Their system is employed to study and analyze hand postures, not the hand motion trajectory as in our system. Tanibata *et al.* [24] introduced off-line method to recognize Japanese Sign Language (JSL) and JSL word in a unconstrained background. Yang *et al.* [48] introduced an ASL

recognition system based on a time-delay neural network. All of the presented works are very inspiring and have different interesting approaches to overcome different problems of sign language recognition. Most of the introduced systems are running in off-line mode, i.e. they collect the feature sequence and start recognition when the gesture has already been performed.

2.3 Gesture Recognition Approaches

Vision based recognition systems use cameras as the input source and are used to interact with the computers. Gestures are tracked from the motion of the hands. Application areas include the interaction with the virtual objects. Other applications include sign language recognition, graphical interface controls, simulation, robot teaching, device control, and virtual reality. In addition, pattern spotting is an important topic of research in speech recognition and computer vision. Moreover, a pattern spotting algorithm is required in order to find predefined patterns in the input data. Major approaches for analyzing and extracting patterns with spatial and temporal variabilities include Neural Network-Based approach (NN) [18, 49, 50, 51], Template Matching-Based approach [12, 34, 52, 53], Hidden Markov Models-Based approach [5, 22, 54] and Conditional Random Fields-Based approach [25, 26, 55].

In these approaches, the features are extracted from the images and then tested against the observed feature set. Specifically, the parameters derived from the hand include contours, edges, image moments, eigenvectors, fingertip etc. Most of these parameters are used as features in the recognition. Two major problems in these approaches are feature selection and training of the data set. Feature selection means how the features are selected for the system to classify correctly and how many features are enough for the system. The second problem is the training of the data set. For training the data, it is always difficult to decide how many samples are enough for training. Classification is then performed on these feature sets. Unlike 3D model based approaches, they can work in real-time because of the extraction of 2D image features. The following sections explore a vision-based analysis of hand gestures with spatio-temporal variabilities.

2.3.1 Neural Network-Based Approach

As large data sets become available, more emphasis is shifted on NN where two approaches of representing temporal information exist. The first is to use a recurrent NN and the second is to use a multilayer feed-forward network, with a sophisticated pre-processing architecture. Although, the neural networks have the ability to represent and recognize static patterns (i.e. postures), they are not suitable for interpreting dynamic gestures (i.e. gestures) [56]. One of main problems which arise in hand gesture spotting is how to model non-gesture patterns.

Vaananen and Boehm [56] have used NN to recognize user gestures and virtual environment visitor which is called Gesture driven Interface in Virtual Environments (GIVEN). The gesture recognition module of GIVEN includes two parts: posture recognition and dynamic gesture recognition. From the DataGlove, the posture recognition part obtains 10 inputs (i.e. two for each finger). After preprocessing, the scaled inputs were fed into feed-forward network, which perform the recognition and send the information to the GIVEN program for further processing. The dynamic gesture recognition part uses back propagation neural network with a sophisticated preprocessing architecture. They have used two sets of input information, ten finger angles and six position information.

Fels [57] has developed Glove-Talk II that translates hand gestures to speech using an adaptive interface. An adaptive interface was to improve the user's performance based on experience with the user. Fels has chosen neural networks because it gives natural models for construction adaptive interfaces as well as the enough speed of running process for real-time control after training. Hand gestures were mapped to allow the hand to serve as an artificial vocal tract, which provides the speech in real-time. The gesture-to-speech process has been divided into consonant and vowel production. Additionally, a gating network has been used to weight the outputs of consonant and vowel network. Different examples from the same user have been employed to train the gating and consonant networks.

Kjeldsen *et al.* [50] have developed a control system for a window based user interface which interacts with the user by visually recognizing hand gestures and performing actions in response. Their approach used two layers architecture: hand tracking and action layers. The hand tracking layer used cheap but coarse techniques to identify and track the user's hand in real-time. The action layer used a grammar to map image events which are identified by the hand-tracking layer to actions of the system. The basis of the grammar was a core cycle which represents three gesture phases: preparation, gesticulation and retraction, and it makes use of both the motion and pose of the hand. Whereas, Stiefelhagen *et al.* [49] used 3D position of head and hands to recognize gestures based on two NNs; one for tilted orientation of a head pose and another for panning. The purpose of using NNs was to process the head's intensity and the disparity where a stereo camera has been employed to capture the data. The combination of gray and depth information achieve good results in contrast to the separate use of gray or depth information.

2.3.2 Template Matching-Based Approach

Gesture models in template matching-based approaches are modeled as a spatio-temporal template. In general, it is difficult to handle template matching for the temporal variability domain because template matching depends on the spatial distance between input data and template. Despite of this difficulty, this approach is

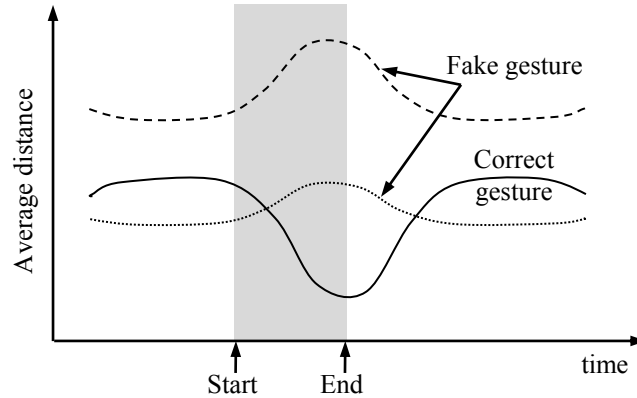


Figure 2.2: Output of CDP matching algorithm. CDP computes the optimal path and the minimum cumulative distance between the gesture models and the input sequence to detect a candidate gesture.

appropriate when the training data set is simple and the variance is small. Waldherr *et al.* [58] introduced a command interface for hand gestures to control the equipped mobile robot. A camera has been considered to track a human and recognize hand gestures, which include arm trajectory. This method achieves promising results when compared to a neural network-based approach. Whereas, Kortenkamp *et al.* [59] proposed a method to recognize hand gestures based on a stereo vision system. This method has the ability to recognize six different gestures like pointing and hand signals. Takahashi *et al.* [12] proposed a Continuous Dynamic Programming (CDP) algorithm in order to segment and recognize meaningful gestures with body and arms. The proposed algorithm used a set of standard sequence patterns to represent meaningful gestures in spatio-temporal form. The input image sequences were compared using CDP matching algorithm for the recognition (Fig. 2.2). The experiments were carried out to choose seven gestures and the results concluded that this model was robust against clothes and background. Seki *et al.* [60] have also used CDP matching for gesture recognition. In their system, the features were based on 2D power spectrum and velocity extraction because the power spectrum of fourier transform and the velocity of arms in images does not depend on parallel translation. They concluded that the features make the gesture recognition system shift-invariant. The drawback of the method is that it does not perform robustly with respect to shape variations.

Alon *et al.* [61] proposed a novel algorithm for gesture spotting and recognition based on CDP. Additionally, they used a pruning method in conjunction with subgesture reasoning process in order to efficiently spot and recognize short gestures. The pruning method has been used to make the system relatively able to estimate small number of hypotheses when compared to CDP. The process of subgesture reasoning

has been employed to alleviate the problem of short gesture when matched to other longer gestures. The experiments showed that the use of this method was faster and 18% more accurate unlike the use of CDP algorithm without any modifications. Alon *et al.* [17] also proposed a unified framework to simultaneously spot and recognize gestures. This framework contains three main processes. Firstly, a spatio-temporal process has been employed to accommodate multiple candidate hand localizations in each frame image. Secondly, a classifier-based pruning process was devised to early refuse weak match patterns in gesture models. Finally, a subgesture reasoning process was built to learn which gesture models could be matched with errors of their parts to other longer gestures. The performance of this framework was the restoration of gesture's occurrence of gestures of interest from a video database which contains continuous gestures in ASL. The experiments showed that the rate of correct detections for digits has been increased tenfold from 8.5% to 85% when compared to CDP method of Oka [62].

Dynamic Time Warping (DTW) is considered as a template-based matching technique, which was used to deal with the problems of temporal variabilities. Moreover, DTW achieves successes in resolving small vocabulary problems. However, the drawback of using DTW is that it requires more templates for representing spatial variabilities during matching process. Another drawback for DTW is associated to the prior selection of the start and the end points of input gestures. This prior selection is not suited for online recognition system because the start and the end points of gestures are not easily inferred. Moreover, recognizing non-gesture patterns is a major problem in case of DTW.

2.3.3 Hidden Markov Models-Based Approach

HMM is one of the best approaches used in pattern recognition as it has the ability to overcome the problems of spatio-temporal variabilities [63]. In addition, HMMs have been successfully applied to gesture recognition, speech recognition and protein modeling etc. [5, 15, 54]. Introduction of HMMs makes the recognition-based segmentation more powerful because segmentation and recognition are optimized simultaneously during recognition with HMMs. Gesture can be divided into two types; a communicative gesture (i.e. key gesture or meaningful gesture) and a noncommunicative gesture (i.e. garbage gesture or transition gesture) [6, 64]. In other words, a nature gesture includes three phases: pre-, key- and post-gesture as shown in Fig. 2.3. The key gesture can be defined as a part of hand trajectory which carries explicit meaning for human. Whereas, pre- and post-gestures represent unintentional movement used to connect key gestures.

Vogler and Metaxas [22] introduced a system based on HMMs and three video cameras to recognize ASL. They used an electromagnetic tracking system to extract 3D parameters of the user's hand and arm. Their system has been carried out in two

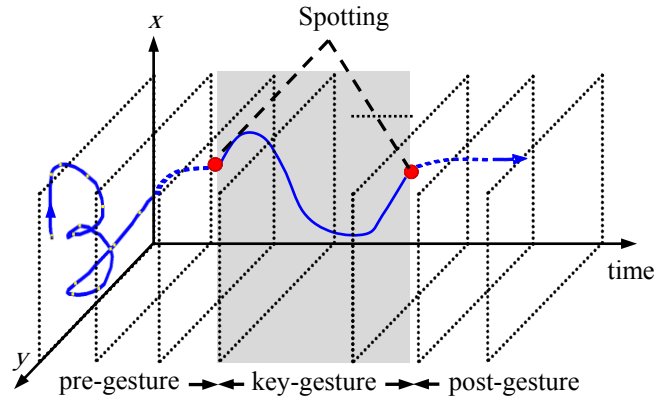


Figure 2.3: Gesture trajectory and spotting with three main phases.

experiments where 99 sentences and 22 signs were tested. The experiments have been performed with isolated signs and continuous sentences. Their system has achieved recognition rates of 94.5% and 84.5% for isolated signs and continuous sentences, respectively. Whereas, Starner *et al.* [5] introduced two HMMs-based systems by using a data set of 40 signs. The first system is depended on the presence of the camera on a desk while the second system plants the camera on a cap worn by the user. The experiments have been carried out on a continuous data set according to hand shapes as extracted features. The systems achieved 92% and 98% accuracies for the first and the second systems, respectively.

Bauer and Kraiss [43] presented HMMs-based system to recognize a German Sign Language (GSL) using colored gloves. The system has been performed with isolated and continuous signs. In their system, subunit HMMs were used to recognize isolated signs and perform the spotting signs. Experiment results demonstrated that the system was successfully recognized spotted hand signs with 81% recognition rate and achieved 92.5% accuracy for 100 isolated signs. Braffort [65] proposed a recognition system for French Sign Language (FSL) in which signs were divided into communicative signs, noncommunicative signs and variable signs. A colored glove was used to extract features like hand appearance and position, which are then employed to HMMs codewords. The experiments were run according to classifiers; one was to recognize communicative signs and the another was to recognize both noncommunicative and variable signs. Their system has achieved 96% recognition rate for vocabulary of seven signs.

The method of Lee and Kim [15] is considered as the first signs of the way dealing with transition gestures as a pattern of separate modeling. This method has been used to address 2D hand trajectory (i.e. gesture trajectory) regardless of taking the hand shapes into account. The drawback of this method is that the number of samples is not considered while merging two states. Kahol *et al.* [33,66] proposed Hierarchical

Activity Segmentation (HAS) algorithm . HAS algorithm used hierarchical layered structure in a dynamic way to represent the anatomy of person. This algorithm has also considered low level parameters of motion to recognize up-bottom motions in addition to conducting numerous attempts to segment a complex person motion sequence (e.g. dancing). The mechanism of this method were subjected to two main steps. In the first step, the boundaries of potential gesture have been recognized with three cues and then employed as second step to naive Bayesian classifier for boundary detection of correct gesture. In order to spot dance sequences, 3D information based on coupled HMMs (cHMMs) were used for individual gesture patterns. The main advantage of this method is that all transition gestures in person motions are considered, unlike other researches which are only interested in key gesture spotting. To spot key gestures exactly, the transition gestures are explicitly modeled. In short, HMMs are capable of modeling spatio-temporal of gestures effectively and can handle non-gesture patterns easily.

2.3.4 Conditional Random Fields-Based Approach

Conditional Random Fields (CRFs) are undirected graphical models that were developed for labeling the sequential data [26]. The key features of CRFs than HMMs are represented in their conditional nature and the dependency assumptions of their computations to ensure tractable inference. HMMs are the generative models which define a joint probability distribution to solve a conditional problem. Moreover, one HMM is constructed per label where HMMs assume that all the observation are independent. Whereas, CRFs overcome the weakness of directed graphical models which suffer from the bias problem as in Maximum Entropy Markov models (MEMMs) [25, 26]. The bias problem is due to the fact that the MEMMs states are locally normalized. There is a difference between HMMs and MEMMs in the calculations for each state. In HMMs, the current observations are based only on the current state, but the current observations for MEMMs depend on the previous and the current states [16]. The difference between CRFs and MEMMs is that CRFs use a single exponential distribution to model all reference labels for a given observation sequence. This means that there is a trade-off for each label according to the weights of each feature function. In MEMMs, each state is employed as exponential model to conditional probabilities of the next state for a given current state. Furthermore, CRFs combine the strength of MEMMs and HMMs on the number of real-world sequence labeling tasks [67, 68].

Yang *et al.* [16] proposed a threshold model based on CRFs which uses an adaptive threshold to spot and recognize gestures in continuous input streams. The experiments were performed with isolated and continuous dataset according to the extracted features. Yang and Sarkar [69] proposed CRFs-based ASL spotting and recognition system using Kanade-Lucas-Tomasi method to extract features from motion trajectory. Their system has the ability to extract and recognize key frames from continuous

sentences. Each key frame has been labeled with sign pattern or non-sign pattern. A major limitation of such methods is that they used the backward spotting technique to first detect the gesture end point. After that, they track back to discover the gesture start point and recognize the segmented gesture.

2.4 Discussion and Conclusion

The aim of this chapter is to present a variety of methods for finding occurrences of patterns in a long input streams. The chapter starts with an overview of the research highlights and the challenges present in the research field from the aspect of three main points: 1) Isolated gesture recognition, 2) Gesture spotting, 3) Sign language spotting and recognition. One of the most challenging tasks associated with the gesture recognition problem is gesture spotting, i.e. the task of detecting the start and the end points of a meaningful gesture (temporal segmentation) to be located in every frame of the sequence (spatial segmentation).

To motivate the gesture spotting problem, the major approaches that include neural network, template matching, HMMs and CRFs are summarized. These approaches gave more attentions to analyze and extract patterns with spatial and temporal variabilities. Two major problems in these approaches are feature selection and training of the data set. Feature selection means how the features are selected for the system to classify correctly and how many features are enough for the system. The second problem is the training data set. For the training data, it is always difficult to decide how many samples are enough for training. Although the neural networks have the ability to represent and recognize static patterns (i.e. postures), they are not suitable for interpreting dynamic gestures (i.e. gestures). One of main problems, which arise in hand gesture spotting is to how model non-gesture patterns. In general, it is difficult to handle template matching for the temporal variability domain because template matching depends on the spatial distance between input data and template. Despite of this difficulty, this approach is appropriate when the training data set is simple and the variance is small. On the other hand, HMMs and CRFs have the capability to deal with the spatio-temporal problem in addition to building a model for the non-gesture patterns with no training data.

These approaches mostly used the backward spotting technique to first detect the gesture end point. Then, they track back to discover the gesture start point and recognize the segmented gesture. Further, there is an inevitable time delay between the meaningful gesture spotting and recognition and this time delay is unacceptable for real-time applications. This chapter is important in the context of understanding the motivation of doing the research and enables to investigate and compare the novel techniques.

Chapter 3

Fundamental Concepts

3.1 Color Models

Color models are mathematical methods used for the synthesis of color spaces and are defined by mixing specific proportions of the three chromatics of the Red (R), Green (G), and Blue (B). The importance of a color is that it helps in object's segmentation, detection and classification. In contrast, gray scale algorithms are sensitive to lighting variations. Image processing systems use different color spaces for different purposes. Color models RGB , YC_bC_r , YUV , HSV and HSI are commonly used in image processing and computer vision applications [70]. In the following sections, color models RGB and YC_bC_r are briefly described.

3.1.1 RGB Color Model

RGB color space is an additive color model in which the three primary colors red, green and blue are mixed together in specific proportions to produce any color. Each of the three primary colors is named a component (i.e. channel) and has an arbitrary intensity in the scene. There is no doubt that the increase in the values of red, green and blue additive primaries increase the amount of these values in the scene. Furthermore, zero intensity for each channel provides black color. When each channel has maximum intensity value, the resulting color is white. When the intensity values of all channels are equal, the resulting color is gray. Moreover, the lighter and darker shades correspond to the lighter/darker gray intensity values 3.1(a)). The basic equation of gray scale intensity I is computed from RGB color intensity as follows;

$$I = \frac{1}{3} \cdot (R + G + B) \quad (3.1)$$

It is easier to obtain a normalized RGB ($nRGB$) from the RGB values using a simple normalization method as follows;

$$r = \frac{R}{R + G + B}, \quad g = \frac{G}{R + G + B}, \quad b = \frac{B}{R + G + B} \quad (3.2)$$

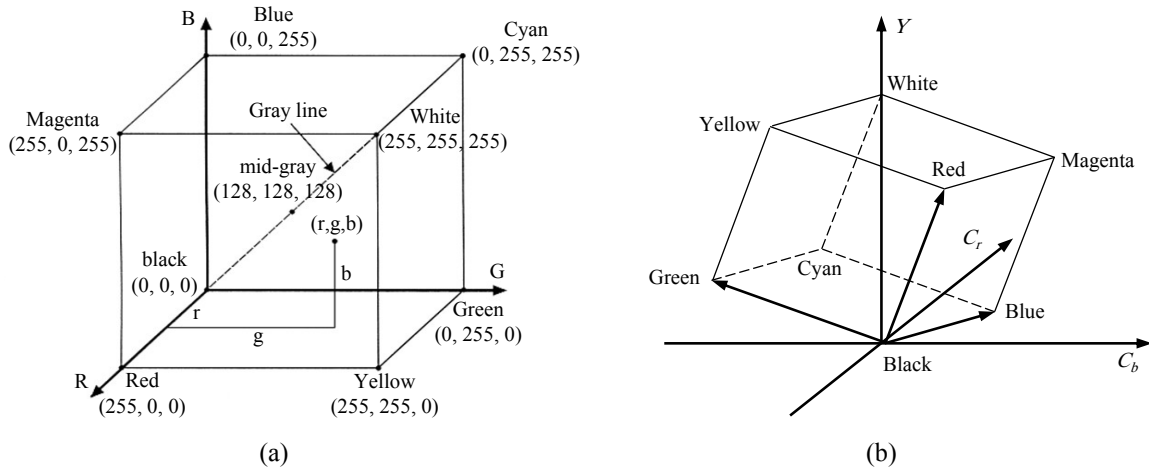


Figure 3.1: (a) RGB color model. (b) YC_bC_r color model.

The third channel in $nRGB$ can be ignored in order to reduce the space dimensionality because it does not hold any significant information. In $nRGB$, summation of the three normalized channels is equal to one ($r + g + b = 1$). The other channels are often named as “pure colors”. By normalization of the source RGB , the color is diminished based on the brightness of r and b . This representation has a better property notably for shiny surfaces. In case of varied light, $nRGB$ is invariant to changes of surface orientation relatively. So, this color space is the most popular among researchers as it can be obtained from RGB easily [71].

3.1.2 YC_bC_r Color Model

YC_bC_r color space is widely used for digital video, video transmission and compression systems. In this format, luminance information contains gray scale intensities and is stored as a single component (Y) while chrominance information is stored as two color-difference components (C_b, C_r). C_b and C_r represent blue and red difference chroma components, respectively (Fig. 3.1(b)). Other color models like YIQ and YUV are widely used for image processing and computer vision applications, and are very similar to YC_bC_r but not identical. YIQ and YUV are used for analog color models while YC_bC_r is a digital color model [72]. In fact, every single pixel in digital RGB color encodings has different R , G and B samples. In contrast to YC_bC_r , the same is not true. In reality, our human eye is more sensitive to variations in the gray scale intensity of a pixel rather than variations in chroma channels. The following equations are used to convert RGB to YC_bC_r color models [73];

$$Y = 0.299R + 0.587G + 0.114B \quad (3.3)$$

$$C_b = 0.564 \cdot (B - Y) \quad (3.4)$$

$$C_r = 0.713 \cdot (R - Y) \quad (3.5)$$

3.2 3D Camera Model

In stereo matching technique, the depth is acquired from a pair of images by the left and right cameras. Analysis and understanding of visible objects based as human eye does is named stereo vision. The purpose of stereo vision is to acquire the depth information through range measurements based on obtained images from cameras with a certain offset [74,75,76]. To get the necessary depth information, a stereo vision module called bumblebee is used (Fig. 3.2(a)), which was developed at Laboratory for Computational Intelligence, University of British Columbia and is being marketed by Point Grey Research¹. It is a challenging problem to estimate the disparity (i.e. determine the corresponding image points) in the stereo vision. This problem is named as corresponding problem. Considering two images which are captured from slightly different views of horizontally displaced cameras, a feature point can be found in the left image I_L at location (x_l, y_l) and in the right image I_R at location (x_r, y_r) . The difference between coordinates of the same features in left and right images is called disparity. Since the cameras are horizontally aligned, only the horizontal displacement is relevant. If the disparity of feature A is different from the disparity of another feature B, their distance to the camera system is different (e.g. point A is closer to the camera than point B when the disparity of feature point A is greater than the disparity of feature point B).

In Fig. 3.2(b), optical axes are normal and parallel to the baseline (b is the baseline which represents the distance between optical centers of the left camera C_L and the right camera C_R) in case of using a normal stereo matching geometry. This, in turn leads to the disparity estimation, which is often applied in the literatures of stereo matching technique [77]. According to Fig. 3.2(b), an object point $P(X, Y, Z)$ is located at (x_r, y_r) in the right image and at (x_l, y_l) in the left image. In addition, the coordinate system of stereo camera is located between the right and the left cameras. Furthermore, values of focal length f and baseline b are positive. The parameter of depth Z that represents the distance between the object point and the baseline, is also positive. Whereas, the coordinates x_l, x_r may be negative or positive. The following equations can be used to measure the relation between the 3-D points with respect to the coordinate systems of the right and the left cameras.

$$X + b/2 = Z \cdot \frac{x_l}{f} \quad (3.6)$$

and similarly for the right camera;

$$X - b/2 = Z \cdot \frac{x_r}{f} \quad (3.7)$$

¹<http://www.ptgrey.com/products/stereo.asp>

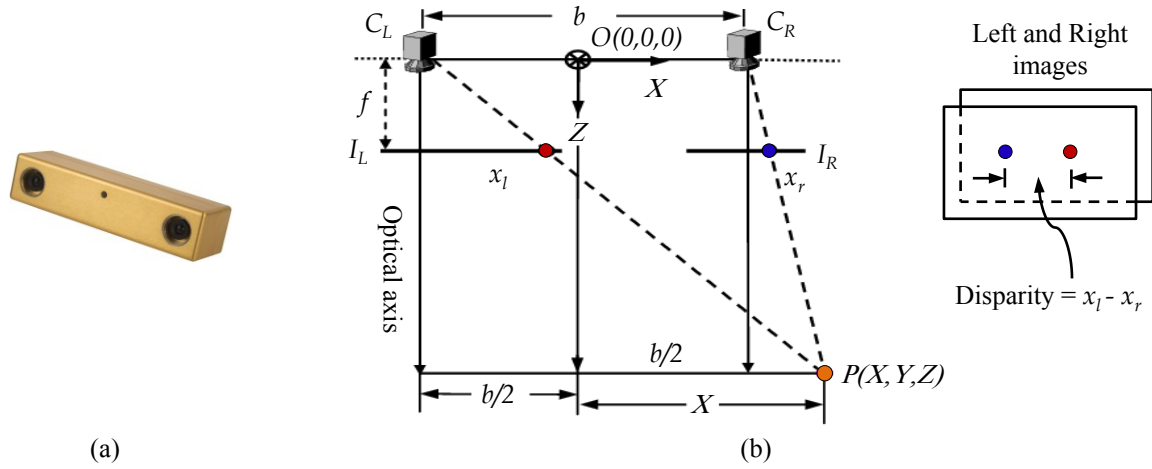


Figure 3.2: (a) Bumblebee stereo vision camera where its size is approximately $160 \times 40 \times 50$ mm and consists of two Sony progressive scan CDDs color sensors with 6mm focal length. (b) The geometry of stereo camera with normal optical axes.

removing X from Eq.3.6 and Eq.3.7;

$$Z = \frac{b \cdot f}{(x_l - x_r)} \quad (3.8)$$

then, the canonical expression relating horizontal disparity $(x_l - x_r)$ to depth Z is obtained as;

$$(x_l - x_r) = \frac{b \cdot f}{Z} \quad (3.9)$$

From a given stereo image pair, the disparity is estimated for each pair of corresponding points. Moreover, this estimation infers 3-D coordinates points of visible scene. There are many methods employed for disparity estimation [78, 79, 80]. These methods differ from one another in many criteria. These criteria are matching primitives, results density, estimation accuracy and implied computation time etc. As 3-D measurement are based on the disparity, so, the accuracy of disparity estimation is an important requirement and necessary to obtain the depth.

Correspondence problem is a big challenge in stereo matching. In addition, the difference in the intensity of corresponding points is large since there is a projective distortion in the occluded boundary. The intensity values of the right and the left images are defined by I_l and I_r respectively. So, the intensity value is computed as;

$$I_l(x, y) = I_r(x + d, y) \quad (3.10)$$

where d represents the disparity function at a pixel position (x, y) .

There are many techniques used to perform the matching criteria as Mean Absolute Difference (MAD) and Normalized Cross Correlation (NCC) [78]. The depth

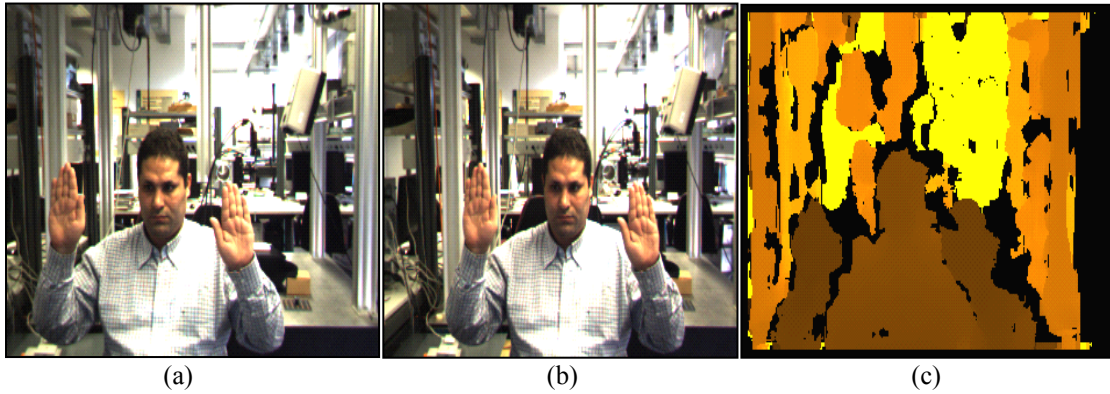


Figure 3.3: (a) Left image of video stream (b) Right image of video stream (c) Depth value of the left and right images.

is estimated in our system by using a MAD matching algorithm which builds the correspondences between the images (Eq. 3.11).

$$MAD(x, y) = \frac{1}{m \cdot n} \cdot \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |M(i, j) - S(i + x, j + y)| \quad (3.11)$$

where $M(i, j)$ is a pixel in reference block that has a dimension $m \times n$ and $S(i, j)$ represents a pixel in search block. x, y are the displacement in x - and y -direction.

In matching tasks, correlation is often employed as an effective similarity measure. Nevertheless, the matching criteria that based on traditional correlation are limited to the short baseline case. The matching algorithm NCC can be employed to estimate the similarity error for each pixel in the image by considering a fixed window in the left image. After that, this window is shifted along the epipolar line in the right image. The correlation between the reference blocks and the search blocks is computed by;

$$NCC(x, y) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} M(i, j) \cdot S(i + x, j + y)}{\sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} M(i, j)^2} \cdot \sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} S(i + x, j + y)^2}} \quad (3.12)$$

Increasing the disparity range increases the time for searching as well as the chances of mismatch; thus it increases the depth range within the image. Decreasing the disparity range allows the system to run faster and decreases the chances of mismatch, and therefore reduces the depth range within the image. The disadvantage of using NCC is that the occlusion boundaries and the results at depth discontinuities are usually unreliable. The reason to use MAD as compared to other (e.g. NCC) approaches is the best optimization potential for speed because MAD method uses less calculations than NCC. Since correspondence is only established at a small number of pixels, the resulting depth map will be very sparse and suitable for on-line applications. The motivation of using the depth information is to define the region of interest in the image

sequence instead of processing whole image. In addition, it resolves the ambiguities between hand and face and identifies the objects under occlusion (Fig. 3.3).

3.3 Segmentation

The human skin is used in image processing research from the human face detection to the hand tracking. The skin color segmentation is the first step which is applied on the captured data after the image acquisition. The purpose of skin color detection is to establish a decision rule which will differentiate between skin and non-skin pixels. When building a system which uses skin color as a feature for hand detection, there are three main difficulties. Firstly, what should be the color space? Secondly, how the skin color distribution should be modeled? The final difficulty is the way of color processing segmentation for the hand sense? There are non-parametric and parametric methods employed for modeling skin color pixels in gesture recognition. The non-parametric methods are the following: Self Organizing Maps (SOM) [81], histogram based techniques and Bayes classifier [82].

The main idea of the non-parametric method is to infer skin color distribution from the training data. Therefore, there is no need to an explicit model for the skin color [71]. Consequently, the non-parametric methods are fast in training. The disadvantage of the non-parametric method is the requirement of much storage space and the potential to generalize the training data. On the other hand, the parametric techniques such as normal Gaussian distribution and Gaussian Mixture Models (GMMs) are based on the modeling of skin distribution. These techniques begin with the modeling of skin and non-skin color using a database of skin and non-skin pixels respectively. GMMs as well as a unimodal Gaussian are employed to estimate the underlying density function. In Gaussian mixture model, a constructive technique is automatically used for estimating the model order. Skin color is a simple but powerful pixel based feature. It allows detection/segmentation of the hands and face in an image. Also, skin color analysis is robust to change in scale, resolution and partial occlusion. The details of these techniques are explained as follow;

3.3.1 Skin Color Modeling Using a Unimodal Gaussian

Segmentation of skin colored regions becomes robust if only the chrominance is used in analysis. Therefore, YC_bC_r color space is used in our work where Y channel represents brightness and C_b, C_r channels refer to chrominance [83]. The channel Y is ignored to reduce the effect of brightness variation and use only chrominance channels to fully represent the color information. Bumblebee stereo camera is used for the input sequence which gives us 2D images along with the depth information. The depth information defines the region of interest (i.e. hands and face regions) in the image which results in the increase of processing speed. Furthermore, the

depth information is used to resolve complex background (i.e. neutralize complex background) completely, as well as illumination variation, and it also increases the accuracy of objects segmentation. Moreover, the skin color region lies in a small region of the chrominance components in YC_bC_r color space (See Fig. A.2 in Appendix A) [84]. So, the distribution of skin color in the chrominance plane is modeled as a unimodal Gaussian. Images are collected which contain human skin pixels as well as non-skin pixels. Therefore, a large database of skin and non-skin pixels is used to train the Gaussian model. Mean and covariance values of the database are used to characterize the model.

Suppose that $x = [C_b; C_r]^T$ represents the chrominance vector of an input pixel. The probability of skin pixel with vector x is calculated as follows;

$$p(x|skin) = \frac{1}{2\pi\sqrt{|\Sigma_s|}} \cdot e^{-\frac{1}{2} \cdot (x-\mu_s)^T \Sigma_s^{-1} (x-\mu_s)} \quad (3.13)$$

where μ_s and Σ_s represent the mean vector and the covariance matrix of s^{th} component respectively. Thus, the mean and covariance which are estimated from the training data, are used to model the skin color distribution as a unimodal Gaussian. The mean and the covariance are formalized as;

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.14)$$

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu) \cdot (x_i - \mu)^T \quad (3.15)$$

where n refers to the number of data points. This model is employed to determine the skin probability image from an input color image (see Section 3.3.3).

3.3.2 Skin Color Modeling Using Gaussian Mixture Models

As described in the previous section, a unimodal Gaussian has been considered for modelling the skin color distribution. The purpose of using a unimodal gaussian is the localization of the skin color according to a small region in the (C_b, C_r) chrominance space. Although, the values of skin color are distributed in a detected region, the histogram of the training data illustrates randomly distributed peaks in this region. Thus, a unimodal Gaussian with a single mean and a single covariance will not give an accurate approximation of the underlying distribution function. On the other hand, a mixture model including a number of Gaussian components do a better approximation in such distributions. So, the mixture models have been developed in order to combine advantages of non-parametric and parametric methods for density estimation [85]. In a given data set, parametric methods are used to estimate the parameters of a standard density function which fits in the given data. Therefore, the density function using parametric techniques is estimated very quickly for new values

of input data. However, the density function using the non-parametric methods can be represented as a linear combination of kernel functions with respect to the center of each kernel on each data point [85]. In general, the non-parametric methods are valid for the forms of density function for the given data. This allows the number of variables to grow partially based on the amount of training data in the model. Thus, the evaluation of density function becomes computationally expensive for new values of input data. According to the skin color modelling using Gaussian mixture, the probability of each color value is a linear combination of their probabilities which are computed from the K Gaussian components. Given a skin color, the probability of a pixel $x = [C_b; C_r]^T$ is as follow;

$$p(x|skin) = \sum_{i=1}^K p(x|i) \cdot p(i) \quad (3.16)$$

where K represents the number of Gaussian components ($K = 3$ in our experiment). To decide the number of components, a method is used to observe the histogram of the dataset in which the selection of K is based on number of peaks for this histogram. In our work, a constructive algorithm which uses the criteria of maximizing a likelihood function is employed to automatically decide the number of components [86]. $p(i)$ is the prior probability of the i^{th} component. It is also called weighting function which is generated from the component i of the mixture. $p(x|i)$ is the Gaussian density model of the i^{th} component.

$$p(x|i) = \frac{1}{2\pi\sqrt{|\Sigma_i|}} \cdot e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)} \quad (3.17)$$

where μ_i and Σ_i represent the mean and the covariance of i^{th} component, respectively.

$$\sum_{i=1}^K p(i) = 1, \quad 0 \leq p(i) \leq 1 \quad (3.18)$$

After deciding the number of components K , the parameters of the mean, covariance and the prior probability for each component are computed from the given dataset. Many approaches have been developed to estimate the parameters of a mixture model for the given dataset [85, 86, 87]. Moreover, these approaches differ from one another in their calculations. One of these approaches is to maximize a likelihood function of the parameters for the given data set [87]. The negative log-likelihood (i.e. an error function E) of the given data set is computed using the following equation;

$$E = - \sum_{n=1}^N \ln \left(\sum_{j=1}^K p(x_n|i) \cdot p(j) \right) \quad (3.19)$$

where N represents the number of data points x_n . Expectation Maximization (EM) algorithm is a special case of Maximum Likelihood (ML) techniques [85, 88] and in this

algorithm, the parameters of mixture model which fits best for the given dataset are estimated for the ML sense. The EM algorithm begins with the initial parameters of Gaussian mixture model and these initial parameters are called ‘old’ parameter values. Then, the values of new parameters are computed using the following equations [88];

$$\mu_j^{new} = \frac{\sum_{n=1}^N p^{old}(j|x_n) \cdot x_n}{\sum_{n=1}^N p^{old}(j|x_n)} \quad (3.20)$$

$$\Sigma_j^{new} = \frac{\sum_{n=1}^N p^{old}(j|x_n) \cdot (x_n - \mu_j^{new}) \cdot (x_n - \mu_j^{new})^T}{\sum_{n=1}^N p^{old}(j|x_n)} \quad (3.21)$$

$$p^{old}(j) = \frac{1}{N} \sum_{n=1}^N p^{old}(j|x_n) \quad (3.22)$$

where

$$p^{old}(j|x_n) = \frac{p^{old}(x_n|j) \cdot p^{old}(j)}{\sum_{i=1}^K p^{old}(x_n|i) \cdot p^{old}(i)} \quad (3.23)$$

The superscript ‘old’ denotes the evaluated quantities using old parameter values. Similarly, the superscript ‘new’ is referred as the computed quantities using old parameters. The parameters of these equations are updated based on the minimization of error function E for the given data set. Therefore, the ‘new’ parameter values become the ‘old’ ones in the next step. This process is iterated until convergence of the error function is reached.

To determine the number of Gaussian components in mixture model of skin data, a cross validation technique is considered [86]. The main idea of this technique is based on the partition of the available data into independent training and validation sets. To minimize the error function, a number of models with different order are examined on the training data set. For each model, the error function is computed for the validation data using EM algorithm. Among these models, one of them with the lowest error is considered as a general model and its order will be optimized in this work.

3.3.3 Skin Probability Image

In the previous sections, the skin color was modeled using either an unimodel or a mixture model Gaussian. The probability of an input pixel representing a skin is computed by these models. According to Bayes formulation [84], the required probability $p(\text{skin}|x)$ is calculated as follows;

$$p(\text{skin}|x) = \frac{p(x|\text{skin}) \cdot p(\text{skin})}{p(x|\text{skin}) \cdot p(\text{skin}) + p(x|\text{non-skin}) \cdot p(\text{non-skin})} \quad (3.24)$$

where the probabilities of *skin* and *non-skin* classes have the same probability value as in Eq. 3.25.

$$p(\text{skin}) = p(\text{non-skin}) = 0.5 \quad (3.25)$$

which provides,

$$p(\text{skin}|x) = \frac{p(x|\text{skin})}{p(x|\text{skin}) + p(x|\text{non-skin})} \quad (3.26)$$

A similar Gaussian model is created for non-skin pixels to determine the probability $p(x|\text{non-skin})$. This model is named the background model. Here, the background is modeled as a unimodal Gaussian in order to reduce the computational complexity for skin probability calculation. Given an input color image, the above ratio and two conditional probabilities are calculated pixel-by-pixel to obtain the probability of each pixel representing skin. Note that, this result in a gray level image where the gray value for each pixel provides the probability of that pixel representing skin. Thus, the skin probability image is determined by the following equation;

$$\text{skin-prob}(i, j) = 255 \cdot p(\text{skin}|x_{ij}) \quad (3.27)$$

where x_{ij} represents the chrominance value of pixel (i, j) .

3.4 Classification

In computer vision, a good choice for classification approaches helps the success of any system and makes it suitable for real-world applications. Classification of symbols in gesture recognition assigns them to a respective class. In this thesis, gestures are handled according to two different classification techniques: generative model such as Hidden Markov Models and discriminative model like Conditional Random Fields.

3.4.1 Hidden Markov Models

The most widely used recognition algorithm for gesture recognition is Hidden Markov Models (HMMs) [63, 89, 90, 91]. Hidden Markov Models are mathematical models of the stochastic process which generates a sequence of observations according to the previously stored information. Statistical approach has many advantages in HMMs like rich mathematical framework, powerful learning and decoding methods, good sequences handling capabilities, flexible topology for the statistical phonology and the syntax. The disadvantages lie in the poor discrimination between the models and in unrealistic assumptions that must be made to construct the HMMs theory, namely the independence of the successive feature frames (i.e. input vectors) and the first order Markov process [92]. The developed algorithms in the statistical framework which uses HMMs are rich and powerful in real-time situations. In addition, Hidden Markov Models are the widely used in practice to implement gesture recognition and understanding systems.

In the Markov chain, every state of the model can only observe a single symbol. However, all states in Hidden Markov Models topology can observe one symbol out of a distinct gesture. The probability of observing a symbol for each state is stored

in the observation probability distribution matrix. For example, the observation probability of a symbol in the state s_1 is considered as the probability to emit the symbol. So, in other words the observation probability distribution is named emission distribution in the recognition task. Furthermore, HMMs states are called hidden for the following reasons. Firstly, the decision of observing a symbol represents the second process. Secondly, the emitter of an HMM only emits the observed symbol. Finally, the emitting states are unknown since the current states are based on the previous states. In the gesture recognition, HMMs are very well known and more flexible due to their stochastic nature.

3.4.1.1 Elements of HMMs

A Hidden Markov Model can be symbolized with $\lambda = (A, B, \pi)$ and is characterized by the following elements [63, 93, 94, 95];

- The set of states $S = \{s_1, s_2, \dots, s_N\}$. N represents the number of states in the model.
- An initial probability distribution for each state π such that;

$$\pi_i = P(s_i), \quad 1 \leq i \leq N \quad (3.28)$$

- An N -by- N transition matrix $A = \{a_{ij}\}$, which is given by;

$$a_{ij} = P(s_j | s_i), \quad 1 \leq i, j \leq N \quad (3.29)$$

where a_{ij} is the probability of the transition from state s_i at time t to s_j at time $t + 1$. The sum of the entries in each row of matrix A must be 1 because it is the sum of the probabilities of making a transition from a given state to each other states.

$$\sum_j a_{ij} = 1 \quad (3.30)$$

- The set of possible emission (an observation) $O = \{o_1, o_2, \dots, o_T\}$ in which T is the length of gesture path.
- The set of discrete symbols $V = \{v_1, v_2, \dots, v_M\}$, where M represents the number of distinct observation symbols per state (i.e. the size of a codeword).
- An N -by- M observation matrix $B = \{b_j(m)\}$, where

$$b_j(m) = P(v_m | s_j), \quad 1 \leq j \leq N, \quad 1 \leq m \leq M \quad (3.31)$$

$$\sum_m b_j(m) = 1 \quad (3.32)$$

where $b_j(m)$ gives the probability of emitting symbol v_m at state s_j . The sum of the entries in each row of matrix B must be 1 for the same previous reason.

In short, a complete specification of the HMMs contains two model parameters (N and M). Additionally, it also includes the observation symbols and the three probabilistic parameters A , B and π . Thus, a compact notation of HMM is as follows;

$$\lambda = P(\pi, A, B) \quad (3.33)$$

Here, λ refers to the parameters set of the model.

3.4.1.2 HMMs Basic Problems

Mathematically, three factors control the use of HMMs. These factors lie in their topologies, the selected features to be emitted and their observation probabilities. The feature selections are based on the observation task. There are three main problems for HMMs; and their solutions helps to employ transitions and observation probabilities in a good way for real-world applications. The problems are:

- **Evaluation problem:** Given the observation sequence O and the model parameter λ , how to compute the probability of observed sequence given the model parameter (i.e. $P(O|\lambda)$)?
- **Decoding problem:** Given the observation sequence O and the model parameter λ , how to determine the best path through λ that generates $O = \{o_1, o_2, \dots, o_T\}$ with maximum likelihood (i.e. best explains the observations)?
- **Estimation problem:** Given the observation sequence O , how to adjust or re-estimate the model $\lambda = P(\pi, A, B)$ to generate $O = \{o_1, o_2, \dots, o_T\}$ with maximum likelihood?

Evaluation Problem

Given the observation sequence O and the model parameter λ , the straight forward method is to calculate $P(O|\lambda)$ through enumerating every possible state sequence s of length T and calculates the corresponding probability $P(O|s_1, s_2, \dots, s_t)$. Suppose that the forward variable $\alpha_t(i)$ is defined as the probability of the partial observation sequence $O = o_1, o_2, \dots, o_t$ at state s_i [63] (Fig. 3.4). Hence,

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, s_i|\lambda) \quad (3.34)$$

It is easy to compute all forward variables α 's at next times using the following recursive relation.

$$\alpha_{t+1}(j) = \sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}), \quad 1 \leq j \leq N, \quad 1 \leq t \leq T - 1 \quad (3.35)$$

where the initial values of α 's are computed as follow;

$$\alpha_1(j) = \pi_j \cdot b_j(o_1), \quad 1 \leq j \leq N \quad (3.36)$$

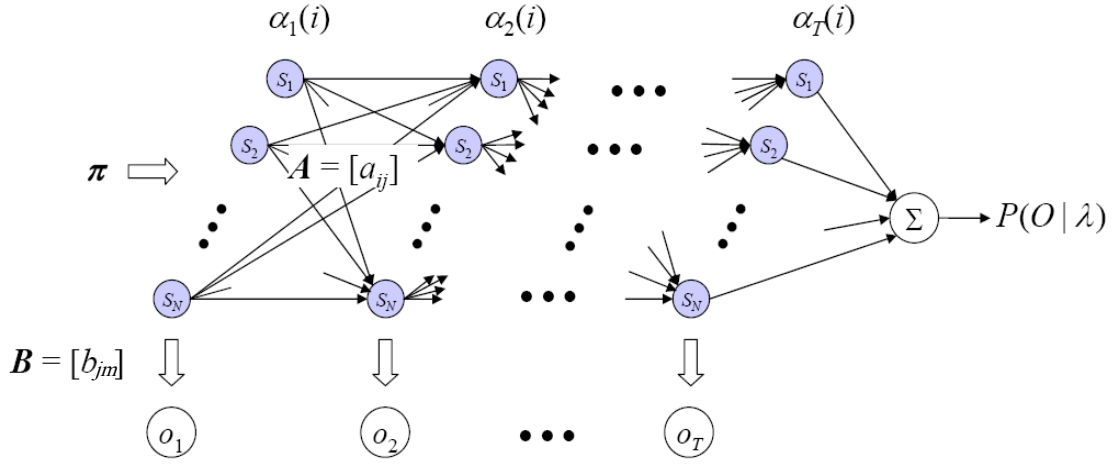


Figure 3.4: Trellis diagram for the forward algorithm.

The procedure is then terminated at T . Thus, the required probability $P(O|\lambda)$ is provided by;

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (3.37)$$

Similarly, the backward variable $\beta_t(i)$ is defined as the probability of the partial observation sequence $o_{t+1}, o_{t+2}, \dots, o_T$ at state s_i . Hence

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T, s_i|\lambda) \quad (3.38)$$

In similar way, the following recursive relationship is used to compute $\beta_t(i)$ as in the calculations of α 's.

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) \cdot a_{ij} \cdot b_j(o_{t+1}), \quad 1 \leq i \leq N, \quad 1 \leq t \leq T-1 \quad (3.39)$$

such that,

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (3.40)$$

Then, the multiplication of α and β for state s_i at time t provides the following estimation.

$$\alpha_t(i) \cdot \beta_t(i) = P(O, s_i|\lambda), \quad 1 \leq i \leq N, \quad 1 \leq t \leq T \quad (3.41)$$

Thereby, this estimation provides another method to compute $P(O|\lambda)$ using both forward α 's and backward β 's variables (Eq. 3.42).

$$P(O|\lambda) = \sum_{i=1}^N \alpha_t(i) \cdot \beta_t(i) \quad (3.42)$$

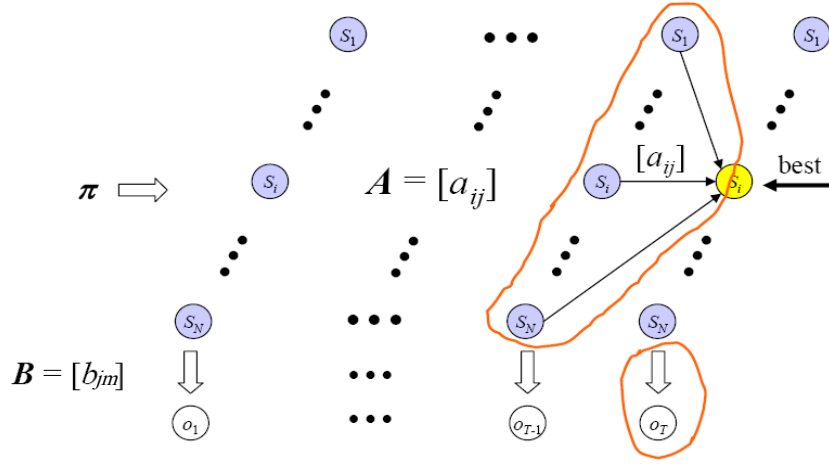


Figure 3.5: Trellis diagram for the forward algorithm, where $\delta_t(j)$ is the highest probability of landing in state j at time t after seeing the observation up to time t .

As a result, the previous equation is very important and more reliable in estimating the required formulas for gradient based training.

Decoding Problem

The motivation behind solving this problem is to explain the best state sequence, which generates the observations $O = o_1, o_2, \dots, o_t$ through model parameter $\lambda = (A, B, \pi)$ with maximum likelihood. For this purpose, Viterbi algorithm is employed [96, 97]. The Viterbi algorithm is a dynamic programming algorithm, which is applied to find the sequence of most likely hidden states to emit the sequence of observed events (Fig. 3.5). The sequence of hidden states is called Viterbi path. The Viterbi algorithm is similar to the implementation of the forward variable $\alpha_t(i)$. The difference is the maximization over the previous states during the recursion step. In order to facilitate the computation, an auxiliary variable is defined as follow;

$$\delta_t(j) = \max\{P(o_1, o_2, \dots, o_t, s_1, s_2, \dots, s_t | \lambda)\} \quad (3.43)$$

The following steps demonstrate how Viterbi algorithm works:

- Initialization: for $1 \leq i \leq N$,
 - a) $\delta_1(i) = \pi_i \cdot b_i(o_1)$
 - b) $\phi_1(i) = 0$
- Recursion: for $2 \leq t \leq T$, $1 \leq j \leq N$,
 - 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 resulted optimal states sequence is $q_1^*, q_2^*, \dots, q_T^*$. $\phi_t(j)$ represents the index of state j at time t , and p^* is the state optimized likelihood function.

Estimation Problem

The training process plays an important role in system performance. To train a HMM, its model parameters are adjusted to obtain the best describe for the observation sequence O_{train} . Until now, there is no analytical solution available for the optimization of HMMs parameters which maximize the probability of observed sequences from training data set [63]. Instead, the Baum-welch (BW) algorithm is used to perform the training process in such a way the $\lambda = (A, B, \pi)$ is optimized with maximum likelihood $P(O|\lambda)$ [98]. BW is a generalized expectation maximization algorithm which is based on the forward and backward variables in its computation [70]. Additionally, this algorithm considers a number of repetitions for the observation sequence to optimize the HMMs parameters. Given a set of observation sequences $o_{train} \in O$, BW calculates the posterior mode estimation and the maximum likelihood estimation for the HMMs parameters (A, B, π) .

The Baum-Welch algorithm is also known as Forward-Backward algorithm. According to the forward and backward variables defined in evaluation problem, two auxiliary variables are defined in order to explain the methodology of BW algorithm. The first variable is the probability of traversing an arc from state i at time t to state j at time $t + 1$ (Fig. 3.6). Mathematically:

$$\xi_t(i, j) = P(s_i \text{ at } t, s_j \text{ at } t + 1 | O, \lambda) \quad (3.44)$$

where ξ represents the transition probability. Moreover, Eq. 3.44 is the same as;

$$\xi_t(i, j) = \frac{P(s_i \text{ at } t, s_j \text{ at } t + 1 | O, \lambda)}{P(O|\lambda)} \quad (3.45)$$

By using forward and backward variables, Eq. 3.45 can be calculated as follows;

$$\xi_t(i, j) = \frac{\alpha_t(i) \cdot a_{ij} \cdot \beta_{t+1}(j) \cdot b_j(o_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) \cdot a_{ij} \cdot \beta_{t+1}(j) \cdot b_j(o_{t+1})} \quad (3.46)$$

The second variable is the state probability (i.e. posteriori probability), which is provided by;

$$\gamma_t(i) = P(s_i \text{ at } t | O, \lambda) \quad (3.47)$$

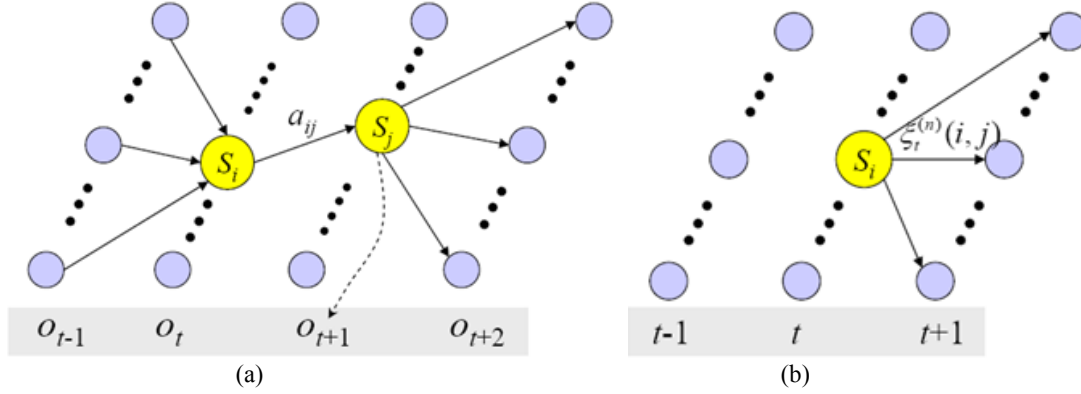


Figure 3.6: Trellis diagram for Baum-Welch learning process. (a) The probability of traversing an arc from state i at time t to state j at time $t + 1$. (b) The probability of state i at time t .

where $\gamma_t(i)$ is the probability of state i at t for given the model parameters and the observation sequence (Fig. 3.6). Similarly, Eq. 3.47 is calculated using forward and backward variables as follows;

$$\gamma_t(i) = \frac{\alpha_t \cdot \beta_{t+1}}{\sum_{i=1}^N \alpha_t \cdot \beta_{t+1}} \quad (3.48)$$

Then, the relationship between $\gamma_t(i)$ and $\xi_t(i, j)$ is provided by;

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j), \quad 1 \leq i \leq N, \quad 1 \leq t \leq M \quad (3.49)$$

Thus, the Baum-Welch algorithm adjusts the new parameters of the HMMs with maximum likelihood of the criterion $P(O|\lambda)$. Given the starting parameters $\lambda = (A, B, \pi)$, the $\hat{\alpha}$ and $\hat{\beta}$ can be computed using the recursive equations of 3.35 and 3.39. After that, the auxiliary variables of $\hat{\xi}$ and $\hat{\gamma}$ are calculated using Eq. 3.46 and Eq. 3.49, respectively. Moreover, the HMMs parameters are updated by using the following equations.

$$\hat{\pi} = \gamma_1(i), \quad 1 \leq i \leq N, \quad (3.50)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^T \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq N, \quad (3.51)$$

$$\hat{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j) \cdot \zeta_{k, o_t}}{\sum_{t=1}^T \gamma_t(j)}, \quad 1 \leq i \leq N, \quad 1 \leq k \leq M, \quad (3.52)$$

where ζ_{k, o_t} is defined as follows;

$$\zeta_{k, o_t} = \begin{cases} 1 & k = o_t \\ 0 & \text{otherwise} \end{cases} \quad (3.53)$$

3.4.1.3 Topologies of HMMs

The choice of HMMs topology has a significant impact in the success of the recognition process as it depends on the available training data and intended model to represent. HMMs have three topologies. The first topology is Ergodic model (Fully Connected model) in which every state of the model can be reached from every other states. Given the fact that every a_{ij} coefficient of Ergodic topology is positive. Fig. 3.7 illustrates this topology for an $N=4$ state model.

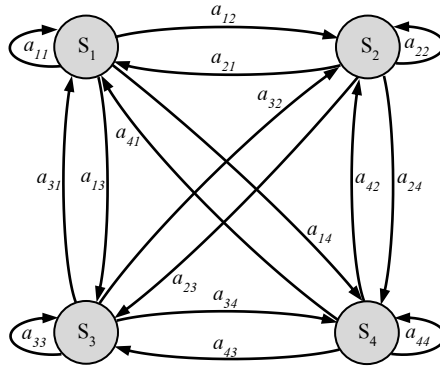


Figure 3.7: Ergodic model with four states.

Hence, for the example of Fig. 3.7, the transition among states has the following form;

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \quad (3.54)$$

The second topology is called Left-Right model or Bakis model [89]. Each state of this model can go to itself or to the following states (Fig. 3.8). In other words, the underlying state sequences for this model have a fundamental property that the state index either stays the same or increases as time increases. As shown in Fig. 3.8, for an $N=4$, the state transition coefficients of this model have the property;

$$a_{ij} = 0, \quad j < i \quad (3.55)$$

The previous equation shows that the state transitions whose indexes are lower than the current state are not allowed. In addition, the initial state probabilities of this model have the property;

$$\pi_i = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad (3.56)$$

Here, the state sequence must begin from s_1 . So, the transition among states for the example of Fig. 3.8 has the form;

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{pmatrix} \quad (3.57)$$

According to the rules of probability theory, the transition coefficients especially for the last state in a left-right model are specified as;

$$a_{NN} = 1, \quad a_{Ni} = 0, \quad i < N \quad (3.58)$$

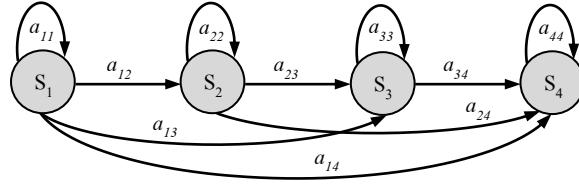


Figure 3.8: Left-Right model with four states.

Another model so-called linear model or Left-Right Banded model (LRB) [93] is illustrated in Fig. 3.9. In LRB model, the states are proceeded from left to right as well as every current state can go back to itself or to the next state with some positive probability. Furthermore, the presented model has an ability to capture variations very well in the temporal extension for the hand gestures introduced in chapter 4. The state transition coefficients of this model (i.e. example of Fig. 3.9) have the property;

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{pmatrix} \quad (3.59)$$

It should be clear that the state transition coefficients for the last state and the other states in a LRB model are specified as;

$$a_{NN} = 1, \quad a_{ij} = 0, \quad i < j; \quad j - i \geq 2 \quad (3.60)$$

Note that, any parameter of HMMs which is initialized with zero value remains zero throughout the re-estimation process. As a result, the Left-Right Banded model will have a negative impact either on the training or the inferencing process.

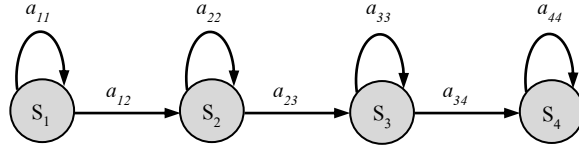


Figure 3.9: Left-Right Banded model with four states.

3.4.2 Conditional Random Fields

Conditional Random Fields (CRFs) are undirected graphical models that were developed for labeling sequential data. CRFs are different than HMMs in their conditional nature and the dependencies assumptions in their computations to ensure tractable inference. In addition, CRFs overcome the weakness of directed graphical models, which suffer from the bias problem as in Maximum Entropy Markov models (MEMMs) [25, 26]. Furthermore, CRFs combine the strength of MEMMs and HMMs on a number of real-world sequence labeling tasks [67]. In our work, each label (state) corresponds to a gesture (e.g. alphabets from A to Z or numbers from 0 to 9). In addition, there is a trade-off for each label according to the weights of each feature function because CRFs use a single exponential distribution to model all reference labels of given observation [16]. The CRFs are satisfied by defining the normalized each product of potential function [99]. In the case of chain-structured CRFs as depicted in Fig. 3.10, each potential function operates on pairs of adjacent label variables y_i and y_{i+1} .

The probability of label sequence y for a given observation sequence x is calculated by;

$$p(y|x, \theta) = \frac{1}{Z(x, \theta)} \cdot \exp \left(\sum_{i=1}^n F_{\theta}(y_{i-1}, y_i, x, i) \right) \quad (3.61)$$

where $Z(x, \theta)$ is the normalized factor given by;

$$Z(x, \theta) = \sum_y \exp \left(\sum_{i=1}^n F_{\theta}(y_{i-1}, y_i, x, i) \right) \quad (3.62)$$

where parameter $\theta = (\lambda_1, \lambda_2, \dots, \lambda_{N_f}; \mu_1, \mu_2, \dots, \mu_{N_g})$, N_f represents the number of transition feature function, N_g refers to the number of state feature function and n is the length of observation sequence x . F_{θ} is defined as follows;

$$F_{\theta}(y_{i-1}, y_i, x, i) = \sum_f \lambda_f t_f(y_{i-1}, y_i, x, i) + \sum_g \mu_g s_g(y_i, x, i) \quad (3.63)$$

where $t_f(y_{i-1}, y_i, x, i) \simeq t_f(y_{i-1}, y_i, x)$ is a transition feature function of the entire observation sequence and labels at positions i and $i - 1$ in the label sequence.

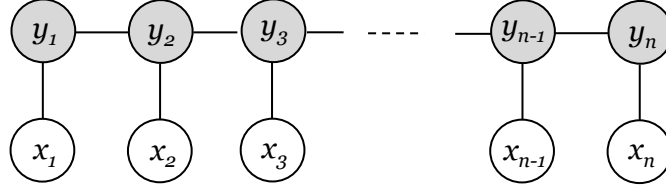


Figure 3.10: Graphical structure of a chain-structured CRFs for sequences. The variables corresponding to unshaded nodes are not generated by the model.

$s_g(y_i, x, i) \simeq s_g(y_i, x)$ refers to a state feature function of the label at position i and the observation sequence. λ_f and μ_g represent the weights of the transition and state feature functions respectively, which can be estimated from training data.

From Eq. 3.61 and Eq. 3.63, the joint probability of a label sequence y given an observation sequence x can be written as follows;

$$p(y|x, \theta) = \frac{1}{Z(x, \theta)} \cdot \exp \left(\sum_{i=1}^n \sum_f \lambda_f t_f(y_{i-1}, y_i, x, i) + \sum_{i=1}^n \sum_g \mu_g s_g(y_i, x, i) \right) \quad (3.64)$$

As CRFs models are similar to HMMs models in their characteristics, it is easy to build a CRFs model by defining a single feature for each label-observation pair (y_b, x) and label-label pair (y_a, y_b) according to the training data set as follow;

$$t_{y_a, y_b}(y_u, y_v, x) = \begin{cases} 1 & \text{if } y_u = y_a \text{ and } y_v = y_b \\ 0 & \text{otherwise} \end{cases} \quad (3.65)$$

$$s_{y_b, x}(y_v, x_v) = \begin{cases} 1 & \text{if } y_v = y_b \text{ and } x_v = x \\ 0 & \text{otherwise} \end{cases} \quad (3.66)$$

Based on the foregoing mentioned, the parameters $\mu_{y_b, x}$ and λ_{y_a, y_b} which corresponds to $s_{y_b, x}(y_v, x_v)$ and $t_{y_a, y_b}(y_u, y_v, x)$ features respectively are equivalent to the logarithms of the HMMs observation and transition probabilities.

3.4.2.1 Learning Parameter for CRFs

The maximum likelihood parameter estimation problem for CRFs which defines the probability distribution (Eq. 3.64) is the task of estimating the parameters $\theta = (\lambda_1, \lambda_2, \dots, \lambda_{N_f}; \mu_1, \mu_2, \dots, \mu_{N_g})$ from training data set $D = \{(x^{(j)}, y^{(j)})\}_{j=1}^{T_d}$. Here, $x^{(j)}$ is an observation sequence of training data set, $y^{(j)}$ represents the corresponding label sequence and T_d refers to the number of training sequences. The learning parameters of CRFs are based on the maximum entropy. According to the principle of maximum

entropy, it is considered a good measure for the variational problems as a finite training data. In addition, it has the ability to justify the probability distribution from incomplete information. The maximization of log-likelihood [99] that learns the parameter θ is computed by²;

$$L(\theta) = \sum_{j=1}^{T_d} \log p(y^{(j)}|x^{(j)}, \theta) = \sum_{j=1}^{T_d} \left(\sum_{i=1}^n F_{\theta}(y_{i-1}^{(j)}, y_i^{(j)}, x^{(j)}, i) - \log Z(x^{(j)}, \theta) \right) \quad (3.67)$$

Up to now, there is no closed solution to Eq. 3.67. Instead, iterative techniques have been used to determine the best solution [16, 26, 99]. Likelihood maximization is performed using a gradient ascent method with Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization technique with 300 iterations to converge [100];

$$\begin{aligned} \frac{\partial L(\theta)}{\partial \theta} = \sum_{j=1}^{T_d} \left(\sum_{i=1}^n \frac{\partial F_{\theta}(y_{i-1}^{(j)}, y_i^{(j)}, x^{(j)}, i)}{\partial \theta} - \right. \\ \left. \sum_x p(y|x^{(j)}) \sum_{i=1}^n \frac{\partial F_{\theta}(y_{i-1}, y_i, x^{(j)}, i)}{\partial \theta} \right) \end{aligned} \quad (3.68)$$

3.4.2.2 Inference CRFs

To compute the probability $p(y|x, \theta)$ of label sequence y for the given new observation sequence x , a set of matrices is computed [26, 99, 101]. To simplify some expressions, special starting y_0 and stopping y_{n+1} states are added. These states are dummy (i.e. observe no symbol and are passed without time delay). Suppose that $p(y|x, \theta)$ is given by Eq. 3.63. For each position i in the observation sequence, $M_i(x)$ is $|\mathcal{Y} \times \mathcal{Y}|$ matrix, which defined as follows;

$$M_i(y', y|x) = \exp(F_{\theta}(y', y, x, i)) \quad (3.69)$$

where $\mathcal{Y} = y_1, y_2, \dots, y_l$ represents a set of labels of the training data set. l refers to the number of the labels, and y', y are the labels of \mathcal{Y} at time i . Using this notation, the conditional probability of a label sequence y given the observation sequence x can be written as the product of the appropriate elements of the $n + 1$ matrices for that pair of sequences (Eq. 3.70);

$$p(y|x, \theta) = \frac{1}{Z(x, \theta)} \cdot \prod_i^{n+1} M_i(y_{i-1}, y_i|x) \quad (3.70)$$

Similarly, the normalization factor $Z(x, \theta)$ for observation sequence x is given by the (*starting, stopping*) entry of the product of all $M_i(x)$ matrices;

$$Z(x, \theta) = \left(\prod_{i=1}^{n+1} M_i(x) \right)_{starting, stopping} \quad (3.71)$$

²More details about the derivation of Eq. 3.67 can be found in [99, 100]

3.4.2.3 CRFs with Hidden Variables

Other approaches including the hidden variables offer several advantages over previous CRFs model. Although the CRFs model the transition among gestures and overcome the weakness of directed graphical models which suffer from bias problem, it does not have the ability to learn the internal sub-structure of gesture sequences. Hidden Conditional Random Fields (HCRFs) are the extension of CRFs, which incorporate hidden state variables to deal well with gesture sub-structure [39, 102]. The main advantage of HCRFs is to automatically model the local interconnection between labels (i.e. states) with hidden variables. However, it cannot model the dynamics among the states (Fig. 3.11(a)).

Latent-Dynamic Conditional Random Fields (LDCRFs) are considered as one of the approaches, which combine the advantages of CRFs and HCRFs by using both extrinsic dynamics and intrinsic sub-structure [103]. The strategy of LDCRFs is based on two main points. Firstly, they learn extrinsic dynamics by modeling the class labels. Secondly, they learn the intrinsic sub-structure of gesture sequence using intermediate hidden states. Thus, LDCRFs models have the ability to overcome the main weaknesses of HCRFs models (Fig. 3.11(b)). LDCRFs models can be used to recognize the un-segmented sequences because they contain a class label per observation. Furthermore, LDCRFs models can efficiently infer the gesture sequences during training and testing processes. HCRFs models have only one label associated to each sequence while CRFs and LDCRFs have one label associated to each time sample in the sequence. As shown in Fig. 3.11, x_j refers to the j^{th} corresponding observation value, h_j is a hidden state that assigned to x_j . y_j represents the label of x_j where the gray circles refer to the observed variables.

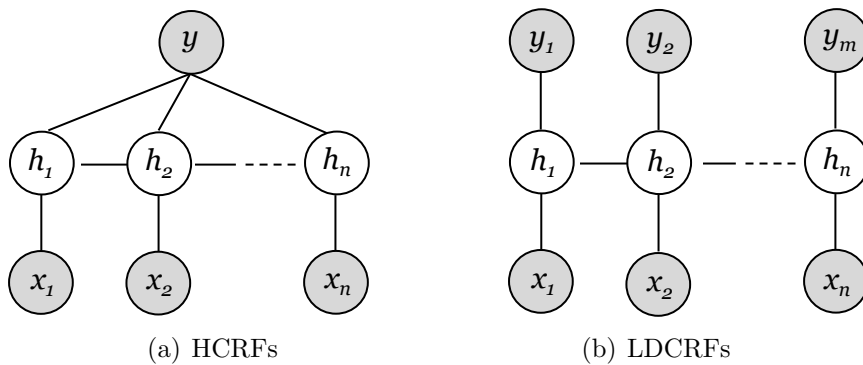


Figure 3.11: Different type of CRFs with hidden states.

3.5 Other Techniques

In this thesis, there are two different techniques which improve the hand gesture recognition. The first one is a relative entropy which reduces the states number

of HMMs topologies. Whereas, k -means algorithm is the second technique and is employed to cluster the extracted features from gesture path.

3.5.1 Relative Entropy

The concept of entropy is to measure the expected uncertainty of a random variable. The entropy function $H(X)$ of a discrete random variable X is determined as follows;

$$H(X) = - \sum_{x \in X} p(x) \cdot \log p(x) \quad (3.72)$$

where $p(x)$ represents a probability of mass function. Here, two probability distributions p and q of random variable $X \in \mathcal{R}$ are considered. The main problem is to measure the difference between these probability distributions. This leads to the idea of the relative entropy of p for given q which was introduced by Kullback [104]. The relative entropy $D(p||q)$ between two probability of mass functions $p(x)$ and $q(x)$ is defined as;

$$D(p||q) = \sum_{x \in X} p(x) \cdot \log \frac{p(x)}{q(x)} \quad (3.73)$$

such that: $p \log \frac{p}{0} = \infty$ and $0 \log \frac{0}{q} = 0$.

So, the relative entropy value is always positive [105] and it has a zero value when $p = q$. In fact, the relative entropy did not reflect the true distance among distributions because it is not symmetric as well as it does not achieve the triangular inequality. Many researchers often consider the relative entropy as the distance among distributions since the distributions ordering can be easily defined and the computations are very simple [106]. Symmetrically, relative entropy is changed slightly as follows;

$$D(p||q) = \frac{1}{2} \sum_{x \in X} \left(p(x) \cdot \log \frac{p(x)}{q(x)} + q(x) \cdot \log \frac{q(x)}{p(x)} \right) \quad (3.74)$$

The relative entropy in Eq. 3.74 finds two states which have the most similar probability distributions. As in our case, the relative entropy provides a way of reducing the number of states for non-gesture model.

3.5.2 Clustering Algorithm

The process of clustering is to classify a given set of patterns into disjoint clusters [107]. To simplify the clustering idea, patterns which lie in the same cluster are alike whereas patterns which belong to two different clusters are dissimilar. There are two goals of clustering algorithms: (1) determining good clusters and (2) doing it efficiently. Clustering has been a widely studied problem in a variety of application domains including data mining and knowledge discovery [108], data compression and vector quantization [109], pattern recognition and pattern classification [110], neural networks, artificial intelligence, and statistics [88].

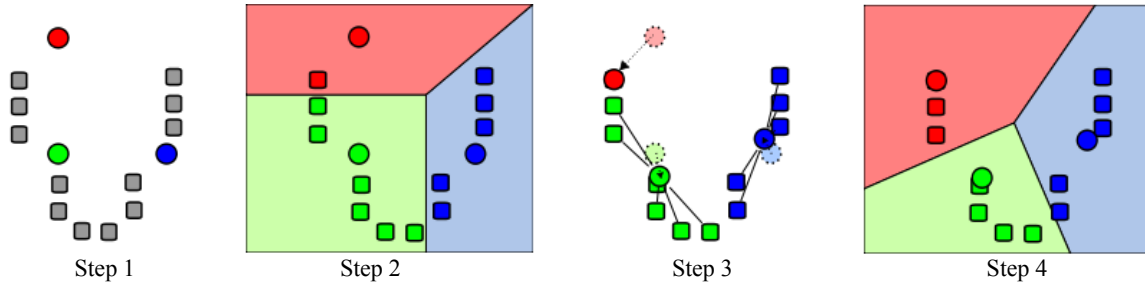


Figure 3.12: Demonstration of k -means clustering algorithm [3].

To represent and formulate the clustering problem, there are many different methods in which the obtained results of clusters or groups are based on the formulated way for each method. For example, the clusters or groups may be exclusive so that each group has its own characteristics which in turn lead to each pattern belongs only to one cluster. On the other hand, the clusters may be overlapping, so, as a consequence, one pattern may be located into different clusters. The clusters can also be probabilistic which mean that a pattern belongs to each cluster is based on the assigned probabilities to it. In addition, the clusters may be hierarchical in which each pattern either assign to a larger cluster or smaller clusters depending on the mechanism used for this method. Although in the literature, there are many different classifications of clustering algorithms as the number of algorithms itself, there is one simple classification so-called k -means algorithm [111], which is used in our work.

The motivation behind using cluster technique like k -means algorithm is to extract more than one feature from hand trajectory. Furthermore, k -means algorithm has many reasons which make it more popular in terms of use such as implementation simplicity, convergence speed, scalability and adaptability to sparse data. Although k -means has the greatest advantage of being easy to implement, it is strongly sensitive to initial points, the quality of the obtained final clusters which depends strongly on the given initial set of clusters. These problems have been addressed well in recent years with significant degrees of success [112].

The main idea of k -means algorithm is very simple and is based on Euclidean distance between all points and center points of clusters. i.e. Firstly, each point is assigned to one of the initialized clusters, then, the cluster center point is recomputed by the mean point on the competent cluster. These processes are iterated until convergence. As shown in Fig. 3.12, k -means algorithm is summarized as follows. Given an initial set of clusters, which may be assigned randomly or by using some heuristic, the k -means algorithm will rotate between the two main steps: *Assignment step* and *Update step*. In assignment step, each instance is assigned to the cluster with the closest mean. While in the update step, the new mean point for each cluster is calculated according to its instance. The following steps are better equipped to

view the k -means in an interesting aspect.

- Step 1: An initial set of clusters K , which are randomly assigned from the data set. In this case $K=3$ with red, green and blue colors.
- Step 2: Each instance is assigned to the cluster with the nearest mean.
- Step 3: The mean of each cluster is replaced with the mean of all instances of its trained vector.
- Step 4: Steps 2 and 3 are repeated until convergence (i.e. there are no changes between two successive iterations).

3.6 Discussion and Conclusion

In this chapter, we have discussed the fundamental techniques which build the basis for understanding this thesis. Color is an important feature which helps in object detection and segmentation. Consequently, different color models were explained in order to demonstrate the different characteristic for each color space and then select the optimal from them for our application. Bumblebee stereo vision camera is used as an input device to capture image sequences. The purpose of the stereo vision camera was to capture the depth information through range measurements based on the obtained images from cameras with a certain offset. The depth information is used to define the region of interest instead of processing whole image, which in turn increases the processing speed. Furthermore, the depth information is used to resolve problems of complex background as well as illumination variation. In case of the overlapping between the hands and face, the depth information is also used to identify the objects under occlusion.

In order to segment hands and face, segmentation technique is exploited which is based on parametric modeling technique. However, the parametric modeling technique depends on skin distribution which includes normal Gaussian distribution and Gaussian Mixture Models. A mixture model consists of a number of Gaussian components and can better approximate such a distribution. Additionally, mixture models combine the advantages of both parametric and non-parametric methods of density estimation. After that, the fundamental formulation of HMMs and CRFs are discussed in details. HMMs are generative models which define the joint probability distribution to solve a conditional problem, thus focusing on modeling the observation to compute the conditional probability. Whereas, CRFs use an undirected graphical model to overcome the weakness of MEMMs. As a part of this opening chapter, relative entropy and k -means algorithm are summarized which are used to improve hand gesture recognition. The relative entropy merges similar probability distributions states and reduces the number of states for a specific model. As the

number of states is reduced, the models inferencing capability is increased and evaluation time is decreased. The motivation behind using k -means algorithm is to extract more than one feature from hand trajectory. Moreover, k -means algorithm is easy and simple to implement, more scalable, converge fast and adaptable to sparse data. The next chapter will explore the isolated hand gesture recognition using HMMs, CRFs, HCRFs and LDCRFs.

Chapter 4

Isolated Hand Gesture Recognition

This chapter proposes a system to recognize the alphabets and the numbers from stereo color image by the motion trajectory of a single hand. In addition, the generative model such as HMMs and the discriminative models like CRFs, HCRFs and LDCRFs are studied to recognize isolated alphabets and numbers. Our system is based on four main stages; automatic segmentation and preprocessing of the hand regions, hand tracking, feature extraction and classification (Fig. 4.1).

In automatic segmentation and preprocessing stage, color and depth information are used to detect hands and face in conjunction with morphological operation. In addition, Gaussian Mixture Models (GMMs) is used for computing the skin probability. For the tracking stage, a robust method in a complex environment is proposed using Mean-shift algorithm in conjunction with depth map. This structure extracts a set of hand postures to track the hand motion with Bumblebee stereo camera as an input device. The depth information computed from stereo camera system is used to identify the region of interest without processing the whole image, which consequently reduces the cost of searching and increases the processing speed. Furthermore, the depth information is used to neutralize completely complex background, as well as illumination variation and it also increases the accuracy of objects segmentation. In case of overlapping between the hands and face, the depth information is also used to identify and separate the objects under occlusion from the rest of image sequences.

Mean-shift analysis uses the gradient of Bhattacharyya coefficient as a similarity function to derive the hand candidate which is mostly similar to a given hand target model. Furthermore, the tracking takes place in the further steps to determine the hand motion trajectory so-called gesture path. In the third stage, combined features of location, orientation and velocity with respect to Cartesian and Polar coordinate systems are computed. Additionally, k -means clustering is employed for HMMs and CRFs codewords.

In the final stage, the isolated hand gesture is handled according to two different classification techniques; HMMs and CRFs to decide which one is the optimal in term of results. HMMs using Ergodic, Left-Right and Left-Right Banded topologies with

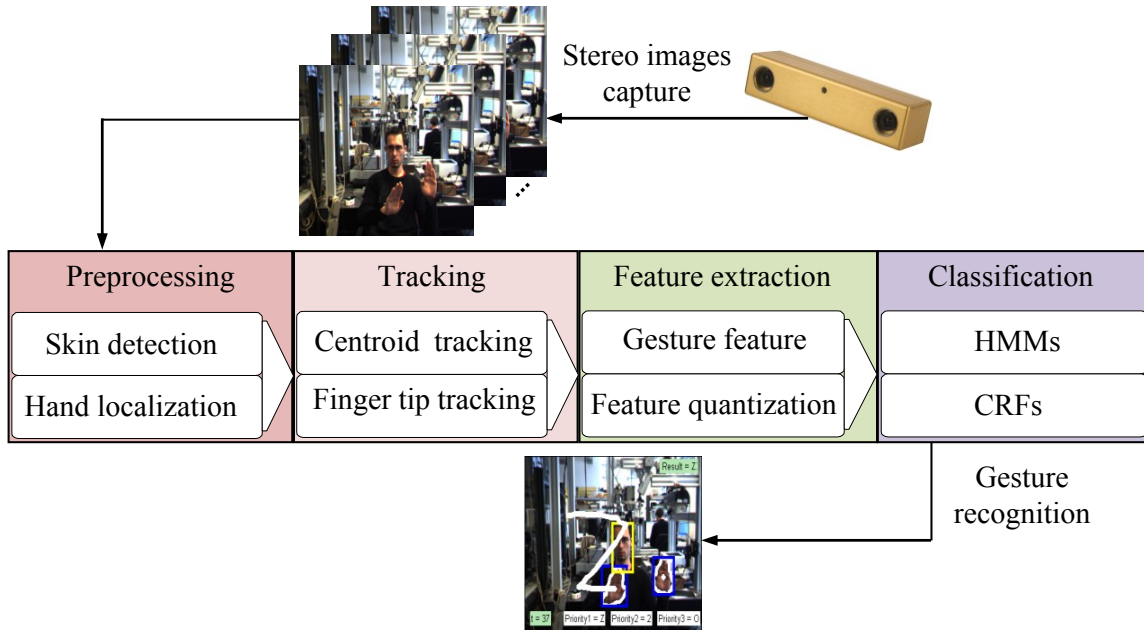


Figure 4.1: Systematic concept of the isolated hand gesture recognition system.

different number of states ranging from 3 to 10 are applied. Additionally, CRFs, HCRFs and LDCRFs with different numbers of window size are applied on combined features of location, orientation and velocity. The proposed system for gesture recognition presents good results under real world conditions with high performance. Image acquisition step is done by Bumblebee stereo camera and contains two set of images, namely 2D images and depth images. In the following sections, components of the proposed gesture system are presented.

4.1 Preprocessing

Our main motivation is to improve the gesture recognition in natural conversations. This requires powerful techniques for skin segmentation and occlusion handling between hands and face to overcome the difficulties of overlapping regions. Therefore, a method for detection and segmentation of the hands in stereo color images with complex background is described in which the hand segmentation and localization takes place using depth map and color information. This stage contains two steps; skin segmentation using GMMs with $YCbCr$ color space and hand localization using blob analysis like *regionprops function*¹ [93,113]. The following sections describe these parts.

¹measures a set of properties for each label region in the label matrix such as area, bounding box and centroid etc. “image processing toolbox of Matlab”

Table 4.1: Gaussian mixture model for skin color database which contains the mean vector, covariance matrix and mixture weight for $K = 3$ clusters.

K	Mean μ	Covariance Σ	Weight
1	(-23.66; 30.01)	$\begin{pmatrix} 23.08 & -24.1 \\ -24.1 & 24.92 \end{pmatrix}$	0.3422
2	(-38.81; 47.36)	$\begin{pmatrix} 23.71 & -16.31 \\ -16.31 & 30.14 \end{pmatrix}$	0.3612
3	(-26.23; 35.29)	$\begin{pmatrix} 57.45 & -17.03 \\ -17.03 & 12.88 \end{pmatrix}$	0.2966

4.1.1 Automatic Segmentation via GMMs

Segmentation of skin colored regions becomes robust if only the chrominance is used in analysis. Therefore, YC_bC_r color space is used in our system where Y channel represents brightness and (C_b, C_r) channels refer to chrominance [91, 94, 114]. The channel Y is ignored in order to reduce the effect of brightness variation and only the chrominance channels are used which fully represent the color information. A large database² of skin and non-skin pixels is used to train the Gaussian model (see Fig. A.1 and Fig. A.2 in Appendix A).

The GMMs technique begins with modeling of skin by using skin database where a variant of k -means clustering algorithm performs the model training to determine the initial configuration of mean vector μ , covariance matrix Σ and mixture weight (Table 4.1). Suppose that $x = [C_b; C_r]^T$ represents the chrominance vector of an input pixel. The probability of skin pixel over vector x for mixture model is a linear combination of its probabilities which is calculated as follows;

$$p(x|skin) = \sum_{i=1}^K p(x|i) \cdot p(i) \quad (4.1)$$

where K is the number of Gaussian components ($K = 3$ in our experiment, because it relies on the skin database used) and is automatically estimated by a constructive algorithm which uses the criteria of maximizing likelihood function [86], $p(x|i)$ is the Gaussian density model of the i^{th} component and $p(i)$ is the mixture weight. It is computed as follows;

$$p(x|i) = \frac{1}{2\pi\sqrt{|\Sigma_i|}} \cdot e^{-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)} \quad (4.2)$$

²18972 skin pixels from 36 different races persons and 88320 non-skin pixels from 84 different images are used to train Gaussian model.

Table 4.2: Unimodel Gaussian for non skin color.

Mean μ	Covariance Σ
$(-19.38; 52.71)$	$\begin{pmatrix} 28.31 & -17.61 \\ -17.61 & 38.20 \end{pmatrix}$

$$\sum_{i=1}^K p(i) = 1, \quad 0 \leq p(i) \leq 1 \quad (4.3)$$

where μ_i and Σ_i represent the mean vector and the covariance matrix of the i^{th} component, respectively.

The expectation maximization algorithm is used to estimate the maximum likelihood of parameters (mean vector, covariance matrix and mixture weight) which run on the training database of skin pixels. For the probability $p(non-skin)$, the non skin color pixels are modeled as a unimodel Gaussian in order to reduce the computational complexity of skin probability (Table 4.2). For more details, the reader can refer to Section 3.3.

4.1.2 Depth Map

Image acquisition step contains 2D image sequences and depth image sequences. For the skin color segmentation of hands and face in stereo color image sequences, an algorithm is devised which calculates the depth value in addition to skin color information. The depth information is gathered by passive stereo measuring based on mean absolute difference and the known calibration data of the cameras. Several clusters are composed from the resulting 3D points. The clustering algorithm is considered as a kind of region growing in 3D which uses two criteria; skin color and Euclidean distance. Furthermore, this method is more robust to the disadvantageous lighting and partial occlusion which occur in real-time environment [115, 116].

The classification of the skin pixels is improved from the top images in Fig. 4.2 by exploiting the depth information which contains the depth value associated with 2D image pixel. The depth information is used to identify the region of interest without processing the whole image which consequently reduces the search cost of a region of interest and increases the processing speed. The depth value lies in the range from minimum depth 30 cm to maximum depth 200 cm in our application. However, the depth range is adaptive according to the region of interest. The values of depth corresponding to the region of interest in the current frame are averaged. Consequently, the depth range according to region of interest is re-calculated in the same way for each subsequent frame. The top images in Fig. 4.2 show the normalized 2D and 3D

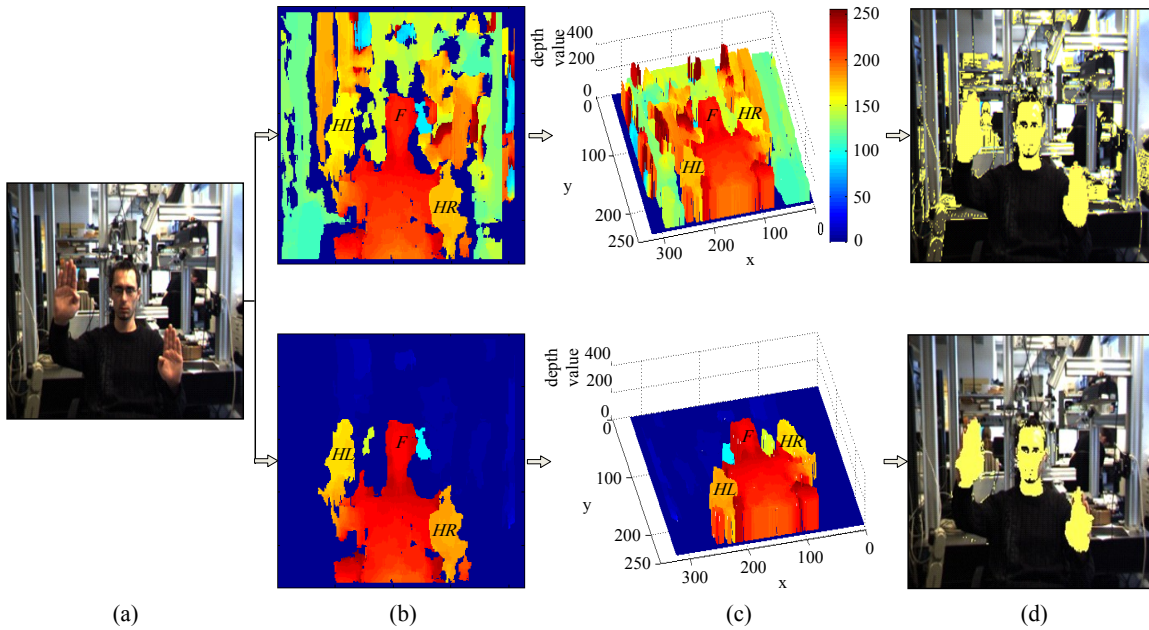


Figure 4.2: (a) Original 2D image. (b) Normalized 2D depth image. (c) Normalized 3D depth. (d) The top image represents skin pixel detection with depth value up to 10 m. In addition, the skin pixel detection without noise is represented in the bottom image (the depth value ranges from 30 cm to 200 cm). Yellow color shows skin pixels detection. F refers to the face, HL and HR represent the left and right hands respectively.

depth image ranges up to 10 m. The normalized depth images are presented for visualization in the range from 0 to 255. Bottom images in Fig. 4.2 show the normalized 2D and 3D depth range of interest (i.e. ranges from 30 cm to 200 cm). It should be noted that the region of interest which includes the hands and face improve skin detection results.

Zero depth image pixels are the pixels having depth value of zero. In some cases, Bumblebee camera does not predict the depth value of pixel and mark its depth as 0 due to the corresponding problem for estimating the disparity (i.e. some pixels of the object are present in one image and are unable to find in the other image). Disparity is defined as the difference between coordinates of the same features in left and right images. This results in a false detection of skin pixels and are marked as non-skin pixels. These depth values are considered irrelevant in the classification of skin pixels. By the given 3D depth map from camera set-up system, the overlapping problem between hands and face is resolved since the hand regions are closer to the camera rather than the face region (Fig. 4.4). Furthermore, the depth information is used to resolve complex background (i.e. neutralize complex background to increase the accuracy of skin segmentation for region of interest) completely.

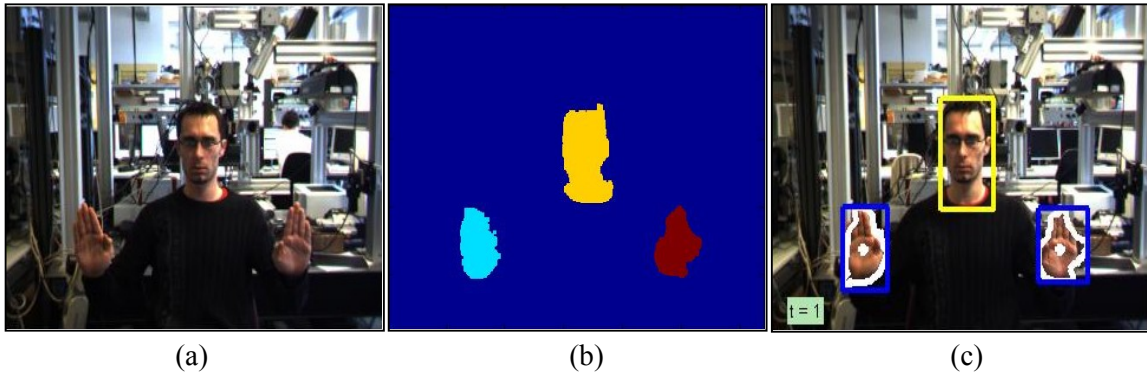


Figure 4.3: Skin color segmentation and hand localization. (a) Source image. (b) Labeled skin detection. (c) Hand localization with a boundary area, bounding box and centroid point.

4.1.3 Hand Localization

For removing the outliers (e.g. noise and spurious components) from the skin probability image, morphological operations (e.g. erosion and dilation) are used because there are small regions which are closer to skin region but does not belong to the human skin. The size and the shape of structuring element used to perform dilation and erosion processes is two-dimensional to probe the input image. Dilation and erosion are used in combination to yield a desired image processing affect. Thereby, the skin color regions are detected (i.e. hands and face). After the labeled skin image is determined (Fig. 4.3(b)), the hands and face are localized using a blob analysis function. This function determines the boundary area (i.e. contour), centroid point and bounding box for each labeled region. Moreover, the contour points are based on a chain code with 8-neighbor connectivity of the segments in a clockwise direction [117]. The area of an object is the summation of all object pixel values. Whereas, the rectangle of ROI is identified using the smallest and the largest x and y coordinates of the localized object. The length of rectangle is the difference between the minimum and the maximum of x coordinates. Similarly, the rectangle width is the difference between minimum and maximum of the y coordinates. With this length and width, the basic features of an object are calculated such as rectangularity, whose measure is invariant to scaling, translation and rotation. Furthermore, the centroid points of detected regions are easily computed by the rectangularity measure.

The next step is the localization of the hands and face and there are four basic criteria to define them. The first criterion is related to x -coordinate values, so that the right contour refers to the right hand, the middle contour is the face and the left contour represents the left hand. The second criterion is the placement of hands and face so that the presence of the face should be in the middle of the screen. Therefore,

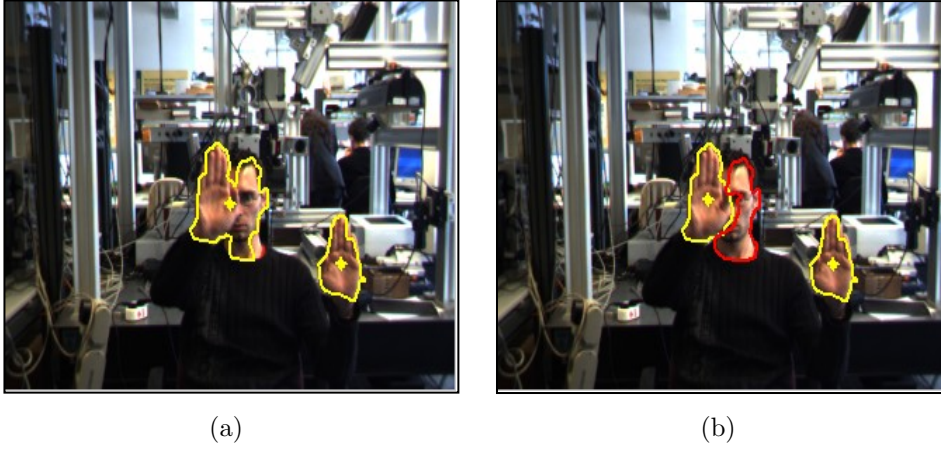


Figure 4.4: Solving overlapping problem between hand and face using depth map. (a) 2D image in which the face and the left hand are occluded. (b) 2D image with labeled hands and face without occlusion.

in this case, it will search only in y -coordinate values for the middle. In the third criterion, the localization of the hands is found by choosing the two small areas and the face represents the big area and the furthest away from the camera. The fourth criterion is to locate the hands and face by assigning them weights relative to the size of their areas. The final detected objects (i.e. hands and face) are illustrated in Fig. 4.3(c). Our attention concentrates on the motion of a single hand in order to obtain the hand trajectory so-called gesture path for a specific alphabet or number. After hand detection, a refinement of the hand description takes place through fingertip detection.

4.1.4 Fingertip Detection

The contour of hand plays a significant role in fingertips detection. At each pixel in hand contour, the neighbor contour points are employed to compute the k -curvature [118, 119]. Here, the curvature is estimated at k , which represents the object boundary point. The main idea is that contour points with high curvature values represent potential peaks which are used as fingertips. The curvature is the ratio between the *length* and the *displacement*. The *length* l is the summation of all distances that a curve has while the *displacement* d is the distance from the first contour point to last contour point. By the following equation, the curvature is computed as follows;

$$k\text{-curvature} = \frac{l}{d} = \frac{\sum_{i=(k-n/2)}^{i=(k+n/2)} \|(P_i - P_{i+1})\|}{\|(P_{k-n/2} - P_{k+n/2})\|} \quad (4.4)$$

where n is the total number of pixels which is used for curvature estimation, P_i and P_{i+1} represent the consecutive points of objects boundary.

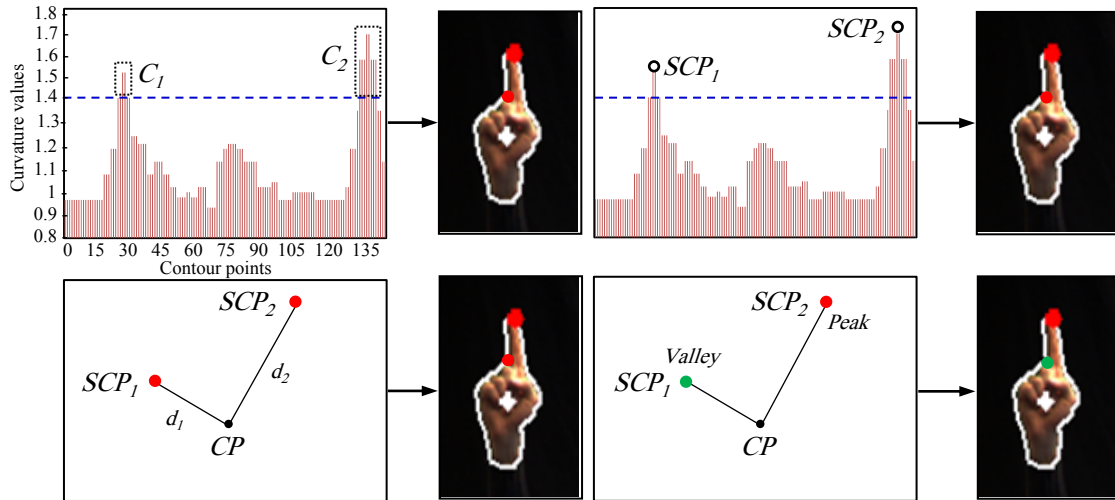


Figure 4.5: Peak and valley detection. In the above graph, maximum local extreme value selects contour points SCP_1 and SCP_2 from the two clusters C_1 and C_2 . The down graph shows that the normalized values greater than 0.5 are detected as fingertip and signed by red point.

The depth map is adaptively set for the objects of interest. For accuracy, the range of depth value is considered from 30 cm to 200 cm (see Section 4.1.2). Moreover, the peaks in hand's contour those curvature values above minimum threshold refer to the fingertips. Empirically, the threshold value is equal to 1.4 in our work. Increasing this threshold value allows for a large number of peaks to be detected. However, reducing this value increases the false positive rate of peaks detection. As illustrated in Fig. 4.5, there are two clusters named as C_1 and C_2 . From these clusters, the maximum value is selected by using maximum local extreme value. As a result, the maximum two points are signed as fingertips (e.g. SCP_1 and SCP_2). Nevertheless, the fingertip can be wrongly detected because this technique considers both peak and valley points as fingertips.

To alleviate this problem, the distance from the center point of an object (CP) to the selected contour points (i.e. SCP_1 and SCP_2) is computed as shown in Fig. 4.5. In addition, the normalized is carried out to scale these points in range of 0 to 1. Thus, the values of points which are greater than 0.5 are classified as fingertips representatives. In the bottom graph of Fig. 4.5, the green point represents a valley whereas the red point represents a fingertip (peak). This technique is the best in term of results for fingertips detection especially in case of using static background (Fig. 4.6(b)). It is because this technique considers the scaling problem to avoid wrong classification between neighboring pixels. In addition, this technique is not costly as compared to other techniques which use histogram analysis to detect fingertip [120], and it works robustly under occlusion because of the depth information.

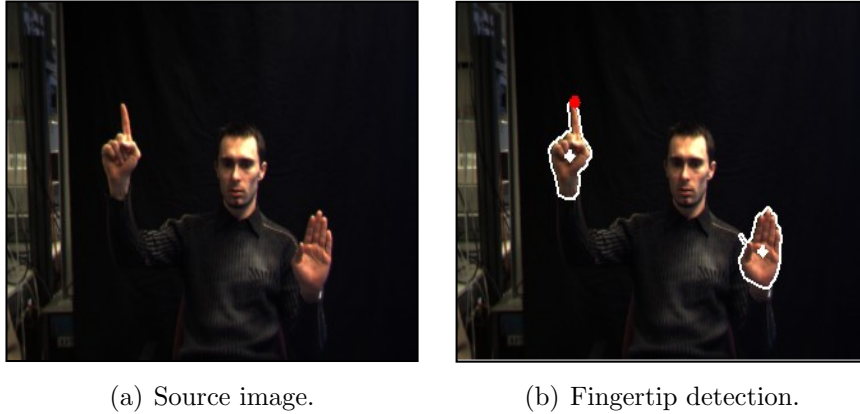


Figure 4.6: Fingertip detection is marked by red point for the left hand and the centroid point is marked by white point.

4.2 Tracking

In this work, a robust method for hand tracking is proposed using Mean-shift analysis in conjunction with depth map. Mean-shift analysis uses the gradient of Bhattacharyya coefficient as a similarity function to derive the candidate of the hand which is mostly similar to a given hand target model. This structure correctly extracts a set of hand postures to track the hand motion. The motivation behind mean-shift analysis is to achieve accurate and robust hand tracking.

4.2.1 Mean-shift Analysis

Mean-shift algorithm is a kernel (i.e. non-parametric) density estimator which optimizes a smooth similarity function to find the direction of the hand target's movement. 16-bin histograms are considered as the representation of the hand's color probability density function (pdf's), as they can satisfy the low-cost requirement for real-time tracking. After localization of the hand's target from the segmentation step, its color histogram is considered with Epanechnikov kernel (monotonic decreasing kernel profile $k(x)$) [121, 122, 123] (Fig.4.7). Epanechnikov kernel assigns smaller weights to pixels farther from the center. Using these weights increase the robustness of the density estimation because the peripheral pixels are the least reliable and are often affected by occlusions.

Let $x_i^*, i = 1, \dots, n$ be the normalized pixel locations in the **hand target model**. The probability of the feature $u = 1, \dots, 16$ in the hand target model histogram is computed as;

$$q_u = F \sum_{i=1}^n k(\|x_i^*\|^2) \delta[b(x_i^*) - u] \quad (4.5)$$

where $b(x_i^*)$ is the index of x_i^* bin in the normalized feature space, δ is the Kronecker

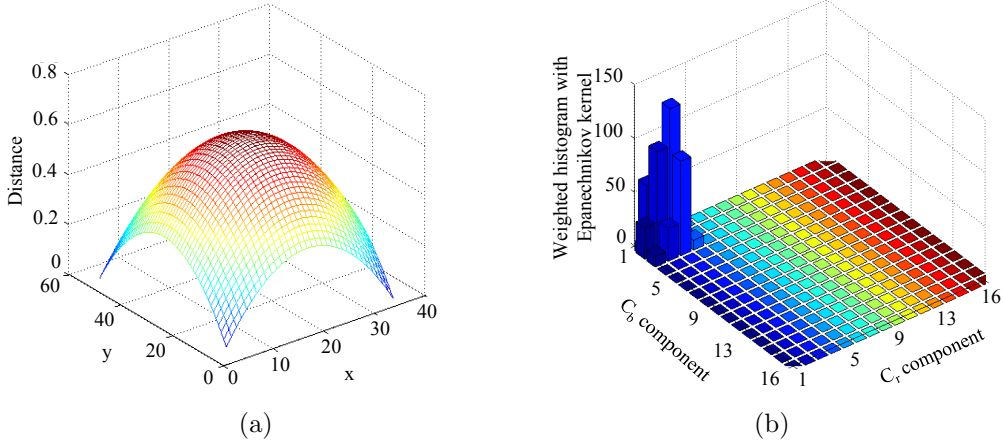


Figure 4.7: Epanechnikov kernel and histogram for the left hand which is depicted in Fig. 4.3. (a) Epanechnikov kernel for the hand target. (b) Projection of 2D weighted histogram of left hand target by using Epanechnikov kernel for (C_b, C_r) components with 16×16 bins.

delta function, equal to 1 only at $b(x_i^*) = u$ and 0 otherwise. The normalization constant F is determined by imposing the condition $\sum_{u=1}^{16} q_u = 1$ where,

$$F = \frac{1}{\sum_{i=1}^n k(\|x_i^*\|^2)} \quad (4.6)$$

For the **hand candidate model** in the next frame, Let $x_i, i = 1, \dots, n_h$ be the normalized pixel locations of the hand candidate which is centered at y . Similarly, the same kernel profile $k(x)$ is used with bandwidth h . The probability of the feature $u = 1 \dots 16$ in hand candidate histogram is calculated as;

$$p_u(y) = F_n \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad (4.7)$$

where the normalization constant F_h is determined as follows;

$$F_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)} \quad (4.8)$$

Moreover, Bhattacharyya coefficient is more suitable to measure the similarity between the hand target model and the chosen candidate. To find the best match of our hand target in the sequential frames, Bhattacharyya coefficient is maximized for Bayes error which arises from the comparison of the target and candidate pdf's. The maximization of Bhattacharyya coefficient between the unit vectors \sqrt{q} and $\sqrt{p(y)}$ which are representing the hand target histogram and hand candidate histogram respectively takes the following form;

$$\rho[p(y_0), q] = \sum_{u=1}^{16} \sqrt{p_u(y_0)q_u} \quad (4.9)$$

It means that the computations need to maximize the term of;

$$\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \quad (4.10)$$

where h is the kernel's smoothing parameter or bandwidth and the weights w_i is derived according to Eq. 4.11.

$$w_i = \sum_{j=1}^{n_h} \sqrt{\frac{q_u}{p_u(y_0)}} \delta[b(x_j) - u] \quad (4.11)$$

The mean-shift procedure is defined recursively and performs the optimization for computing the mean-shift vector. In short, mean-shift iteration uses the gradient of similarity function as an indicator of the direction of hand's movement (Eq. 4.12).

$$y = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i} \quad (4.12)$$

Since the scale of the hand candidate often changes in time, the bandwidth h of the kernel profile in Eq. 4.7 has been adapted accordingly. The bandwidth is measured in the current frame by running the hand candidate localization by three times with small fractions of ± 0.1 . The best yield of hand candidate localization is obtained according to the largest Bhattacharyya coefficient.

There are three types of occluded problems which should be taken into account during generation of hand trajectory. The problems are:

- hand-hand occlusion: this problem is occurred when the two hands overlap each other during the motion, and this in turn leads to loss of real explanation for the gesture path to infer wrong features.
- hand-face occlusion: this problem is occurred when the hand overlaps with the face and vice versa.
- hand-face-hand occlusion: this problem is a far greater challenge in terms of the implementation process when overlapping is occurred between the two hands and face at the same time during the motion.

The problems faced by the system are simple as they only rely on the generation of left hand trajectory. Hands are often in front of the face. So, hand-face occlusion problem is more frequent in our system and therefore overcome by using depth information. The depth information is important for at least two reasons. First, if some features of hand exist in the background (for instance, background represents here the face), their relevance of hand localization is diminished. Second, it is difficult to exactly localize hand because of improper use of background features which makes the similarity measure impossible to identify the appropriate hand target scale. The other two

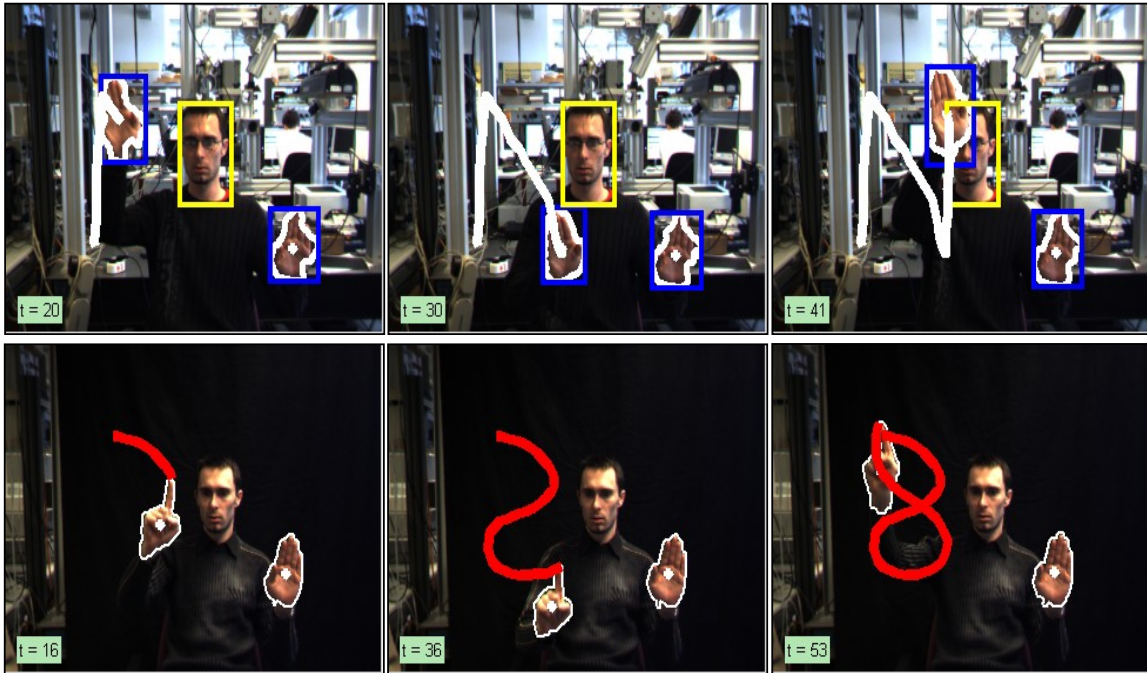


Figure 4.8: Hand gesture path for alphabet ‘N’ using the centroid point and number ‘8’ using fingertip detection.

problems (i.e. hand-hand occlusion and hand-face-hand occlusion) are solved by using mean-shift algorithm in conjunction with Kalman filter [121,123], which will be taking in account in future. In proposed system, mean-shift algorithm is used with the help of depth map to retrieve the extracted features during occlusion. The hand gesture path is obtained by finding the correspondences of detected hand between successive images (Fig. 4.8). Fig. A.6 in appendix A shows successful tracking in presence of partial occlusion and overlapping between hands and face. In addition, the number of mean-shift iteration is 1.61 per frame for both left and right hands, which in turn makes the system robust and capable for real-time implementation.

4.2.2 Trajectory Smoothing

The hand gesture path is determined either by connecting the centroid points or by fingertip detection as described in the previous sections. The input images are usually unstable due to change in illumination conditions, cluttered backgrounds and shaking while moving. So, it will cause frequent, sharp changes in the centroid or fingertip points. In order to efficiently overcome these unexpected changes, the trajectory points are smoothed (i.e. the mean values of a specified point with its neighbors points) using Eq. 4.13. Consequently, the obtained gesture path represents a set of the points in a spatio-temporal space as described in Eq. 4.14. Fig. 4.9(a) shows an

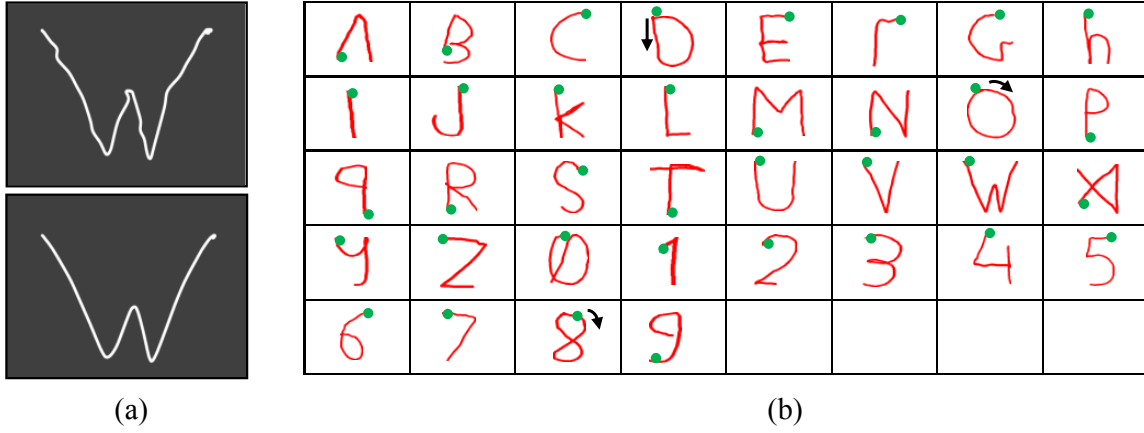


Figure 4.9: (a) Smoothing result for gesture path ‘W’, where the above curve refers to original trajectory and the down curve represents a smoothed trajectory. (b) Hand gesture path shapes for alphabets (A-Z) and numbers (0-9). Green points denote the start points of gesture path explaining the trend.

example of the results of smoothing for gesture path ‘W’.

$$(\hat{x}_t, \hat{y}_t) = \left(\frac{x_{t-1} + x_t + x_{t+1}}{3}, \frac{y_{t-1} + y_t + y_{t+1}}{3} \right) \quad (4.13)$$

$$\text{Gesture path} = \{(\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), \dots, (\hat{x}_t, \hat{y}_t), \dots, (\hat{x}_T, \hat{y}_T)\} \quad (4.14)$$

where (x_t, y_t) refers to the centroid or fingertip point at time t and T is the length of the hand gesture path. The hand gesture paths for alphabets (A-Z) and numbers (0-9) are depicted in Fig. 4.9(b).

4.3 Feature Extraction

Selection of good features for the recognition of the hand gesture path plays a significant role in system performance. There are three basic features: location, orientation and velocity. In the next subsections, the effectiveness of extracted features from spatio-temporal hand gesture path are analyzed according to two categories; features in Cartesian space (x, y) and features in Polar space (ρ, φ) , to decide the optimal in term of results. Additionally, the combination among these features are studied to test their recognition rate.

4.3.1 Features Analysis in Cartesian Space

A gesture path is spatio-temporal pattern that consists of hand centroid points (x_{hand}, y_{hand}) . The coordinates in the Cartesian space can be extracted from gesture frames directly. For this purpose, two types of location features are considered.

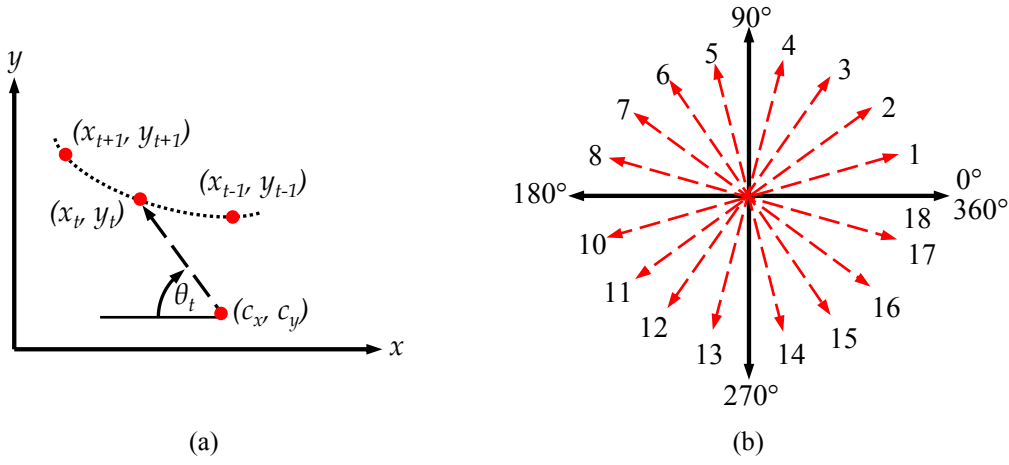


Figure 4.10: (a) Orientation according to the centroid of gesture path. (b) The directional codewords from 1 to 18 in case of dividing the orientation by 20° .

The first location feature is Lc which measures the distance from the centroid point to all points of gesture path because different location features are generated for the same gesture according to different starting points (Eq. 4.15). The second location feature is Lsc which is computed from the start point to the current point of gesture path (Eq. 4.17).

$$Lc_t = \sqrt{(x_{t+1} - C_x)^2 + (y_{t+1} - C_y)^2} \quad (4.15)$$

$$(C_x, C_y) = \frac{1}{n} \left(\sum_{t=1}^n x_t, \sum_{t=1}^n y_t \right) \quad (4.16)$$

$$Lsc_t = \sqrt{(x_{t+1} - x_1)^2 + (y_{t+1} - y_1)^2} \quad (4.17)$$

where, $t = 1, 2, \dots, T - 1$ and T represents the length of hand gesture path. (C_x, C_y) refers to the centroid of gravity at n points. To verify the real-time implementation, the centroid point of gesture path is computed after each frame.

The second basic feature is the orientation which gives the direction of the hand when traverses in space during the gesture making process. As described above, orientation feature is based on the calculation of the hand displacement vector at every point which is represented by the orientation according to the centroid of gesture path (θ_{1t}), the orientation between two consecutive points (θ_{2t}) and the orientation between start and current gesture point (θ_{3t}) (Fig. 4.10).

$$\theta_{1t} = \tan^{-1} \left(\frac{y_{t+1} - C_y}{x_{t+1} - C_x} \right), \theta_{2t} = \tan^{-1} \left(\frac{y_{t+1} - y_t}{x_{t+1} - x_t} \right), \theta_{3t} = \tan^{-1} \left(\frac{y_{t+1} - y_1}{x_{t+1} - x_1} \right) \quad (4.18)$$

The third basic feature is velocity which plays an important role during gesture recognition phase particularly at some critical situations. The velocity is based on the

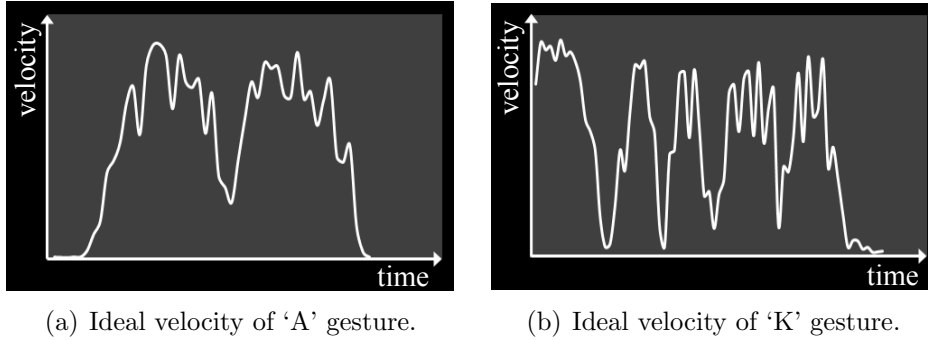


Figure 4.11: Differences in velocity of gesture 'A' and gesture 'K'.

fact that each individual hand gesture is constructed at different speeds, such that the velocity of hand decreases at the corner points of gesture path. For example, the simple gesture 'A' has an almost non-varying speed while a complex gesture 'K' has varying speeds during gesture generation (Fig. 4.11). The velocity is calculated as Euclidean distance between the two successive points divided by the time t (i.e. in terms of the number of video frames) as follows;

$$V_t = \sqrt{\left(\frac{x_{t+1} - x_t}{t}\right)^2 + \left(\frac{y_{t+1} - y_t}{t}\right)^2} \quad (4.19)$$

In the Cartesian coordinate system, different combination of features is used to obtain a variety of feature vectors. For example, the feature vector at frame $t + 1$ is obtained by union of locations features (Lc_t, Lsc_t), locations features with velocity feature (Lc_t, Lsc_t, V_t), orientations features ($\theta_{1t}, \theta_{2t}, \theta_{3t}$), orientations features with velocity feature ($\theta_{1t}, \theta_{2t}, \theta_{3t}, V_t$) and locations features with orientations features and velocity feature ($Lc_t, Lsc_t, \theta_{1t}, \theta_{2t}, \theta_{3t}, V_t$).

Each frame contains a set of feature vectors at time t where the dimension of space is proportional to the size of feature vectors. In this manner, gesture is represented as an ordered sequence of feature vectors, which are projected and clustered in space dimension to obtain discrete codeword and are used as an input to HMMs. This is done using k -means clustering algorithm [124, 125, 126, 127], which classifies the gesture pattern into K clusters in the feature space.

4.3.2 Features Analysis in Polar Space

Polar coordinate is directly calculated from the Cartesian coordinates which are generated from hand gesture path. To obtain the normalized polar coordinates, we use the radius from center point of gesture path (Eq. 4.21) and the radius between the start and the current gesture point (Eq. 4.23).

$$rc_{max} = \max(Lc_t), \quad \rho_{ct} = \frac{Lc_t}{rc_{max}}, \quad \varphi_{ct} = \frac{\theta_{1t}}{2\pi} \quad (4.20)$$

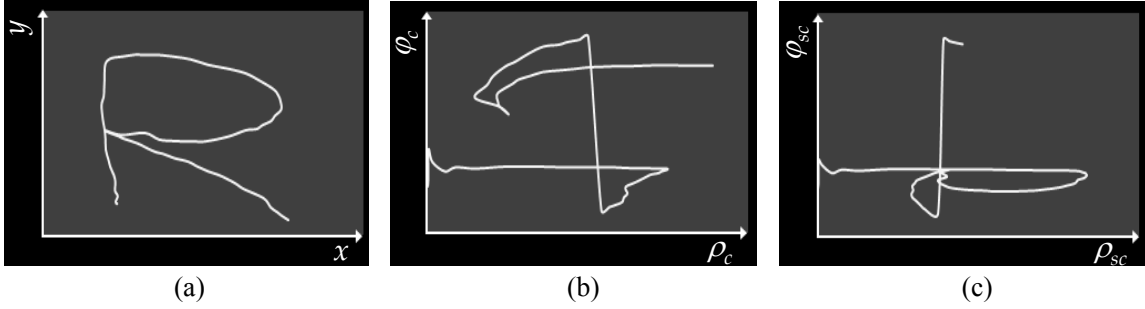


Figure 4.12: Transformation of gesture path ‘R’ from Cartesian to Polar coordinate spaces. (a) x - y space of gesture ‘R’. (b) ρ_c - φ_c space of gesture ‘R’. (c) ρ_{sc} - φ_{sc} space of gesture ‘R’.

$$F_c = \{(\rho_{c1}, \varphi_{c1}), (\rho_{c2}, \varphi_{c2}), \dots, (\rho_{cT-1}, \varphi_{cT-1})\} \quad (4.21)$$

$$rsc_{max} = \max(Lsc_t), \quad \rho_{sct} = \frac{Lsc_t}{rsc_{max}}, \quad \varphi_{sct} = \frac{\theta_{3t}}{2\pi} \quad (4.22)$$

$$F_{sc} = \{(\rho_{sc1}, \varphi_{sc1}), (\rho_{sc2}, \varphi_{sc2}), \dots, (\rho_{scT-1}, \varphi_{scT-1})\} \quad (4.23)$$

where rc_{max} is the longest distance from the center point to each point of hand trajectory at frame $t + 1$ and rsc_{max} represents the longest distance from the start point to each point in the hand gesture path (Eq. 4.22).

In polar space, different combination of features are used to obtain a variety of feature vectors. For example, feature vector at frame $t + 1$ is obtained by union of locations features from the centroid point with velocity feature $(\rho_{ct}, \varphi_{ct}, V_t)$, locations features from the start and the current point with velocity feature $(\rho_{sct}, \varphi_{sct}, V_t)$, and a combination of all $(\rho_{ct}, \varphi_{ct}, \rho_{sct}, \varphi_{sct}, V_t)$. Figure 4.12 shows the representation of the same gesture ‘R’ according to x - y , ρ_c - φ_c and ρ_{sc} - φ_{sc} spaces, respectively. It is observed that there is an obvious variance in the representation of gesture ‘R’ especially in ρ_c - φ_c and ρ_{sc} - φ_{sc} . This variance is important in order to find influential features for the suggested system.

4.3.3 Vector Normalization and Quantization

The extracted features are normalized or quantized to obtain the discrete symbols which are used as an input to HMMs and CRFs. The basic features such as location and velocity are normalized with different scalar values (*Scal.*) ranging from 10 to 30 when used separately. The scalar values increase the robustness for selecting the normalized feature values. The normalization is done as follows;

$$Norm_{max} = \max_{i=1}^{T-1}(Norm_i) \quad (4.24)$$

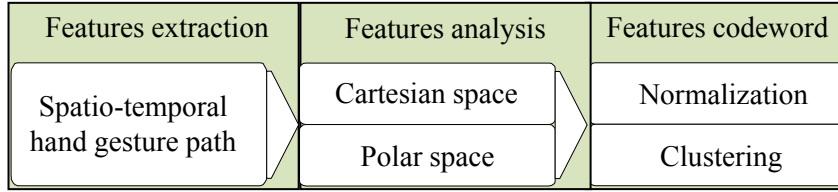


Figure 4.13: Simplified structure shows the main processes for feature extraction stage of isolated gesture recognition system.

where $Norm_i$ represents the feature vector of dimension i to be normalized and $Norm_{max}$ is the maximum value of the feature vector which is determined from all the T points in the gesture trajectory.

$$Fnorm_i = \frac{Norm_i}{Norm_{max}} \cdot Scal. \quad (4.25)$$

According to Eq. 4.25, the normalized value of the feature vector $Fnorm_i$ is computed to obtain feature codes which lie between 10 to 30. The normalization of orientation features is studied with different ranges for codewords to decide the optimal range. Moreover, the normalization of the orientation features is estimated by dividing them by 10° , 20° , 30° and 40° to obtain their codewords which are employed for HMMs and CRFs. The main processes of feature extraction stage according to Cartesian and Polar coordinate systems are illustrated in Fig. 4.13.

On our combined features (i.e. in Cartesian and Polar coordinate systems) as described in pervious sections, k -mean clustering algorithm is used to classify the gesture feature into K clusters on the feature space. The motivation behind using k -means algorithm dues to the ease of representation, more scalable, converge faster and adaptable to sparse data. In addition, more than one feature is extracted from hand trajectory so that they are quantified into a discrete vector which is used as an input to HMMs and CRFs. k -mean algorithm is based on the minimum distance between the center of each cluster and the feature point [30, 128]. The set of feature vectors is divided into set of clusters. This allows us to model the hand trajectory in the feature space by different clusters. The calculated cluster index is used as an input (i.e. observation symbol) to HMMs and CRFs. However, the best number of clusters in the data set is usually unknown.

In order to specify the number of clusters K for each execution of k -means algorithm, the values of $K = 28, 29, \dots, 37$ are considered and studied to decide the optimal in terms of their impact on gesture recognition. Theoretical, cluster number approximately ranges from 28 to 37, so it depends on the numbers of segmented parts in alphabets from A to Z and numbers from 0 to 9; however, each straight-line segment is classified into a single cluster.

Suppose there is a trained data set, which contains n feature vectors x_1, x_2, \dots, x_n such that all these vectors are from the same class. The number of clusters is k with $k < n$ condition. Let the mean of feature vectors that belong to cluster i is symbolized by m_i . Furthermore, the minimum distance classifier is employed to efficiently separate the cluster. Note that, vector x belongs to cluster i if $\|x - m_i\|$ represents the minimum distance as compared to its other k distances. The following procedure shows that how k -means algorithm works.

Input: Given a sample set of vectors and the desired Codebook size of k
Output: Determine the update Vector Quantization Codebook

Build up randomly an initial Vector Quantization Codebook for the means m_1, m_2, \dots, m_k

```

while there are changes between two successive iterations do
  Use the estimated means to classify each sample of train vectors into one of
  the clusters  $m_i$ 
  for  $i = 1$  to  $k$  do
    Replace  $m_i$  with the mean of all samples of the trained vector for
    cluster  $i$ 
  end
end

```

Algorithm 1: k -means clustering algorithm for Vector quantization

4.4 Classification

In the proposed system, classification is the last stage of the work. Classification of the symbols in gesture recognition assigns them to respective classes. Throughout this stage, the isolated hand gesture is handled according to two different classification techniques HMMs and CRFs to decide which one is the best in terms of performance. The following two sections discuss how HMMs and CRFs are employed for the classification of alphabets and numbers.

4.4.1 Classification Using HMMs

Baum-Welch algorithm plays a significant role in the suggested system for gesture classification where it is used for training of the initialized HMMs parameters $\lambda = (A, B, \pi)$. The gesture recognition module matches the tested gesture against database of reference gestures to classify it in the class where it belongs to. Thereby, the hand gesture path is recognized corresponding to the maximal likelihood of all gestures models using Viterbi algorithm. The maximal gesture is defined as a gesture which has the largest value among all the gestures models (Fig. 4.14).

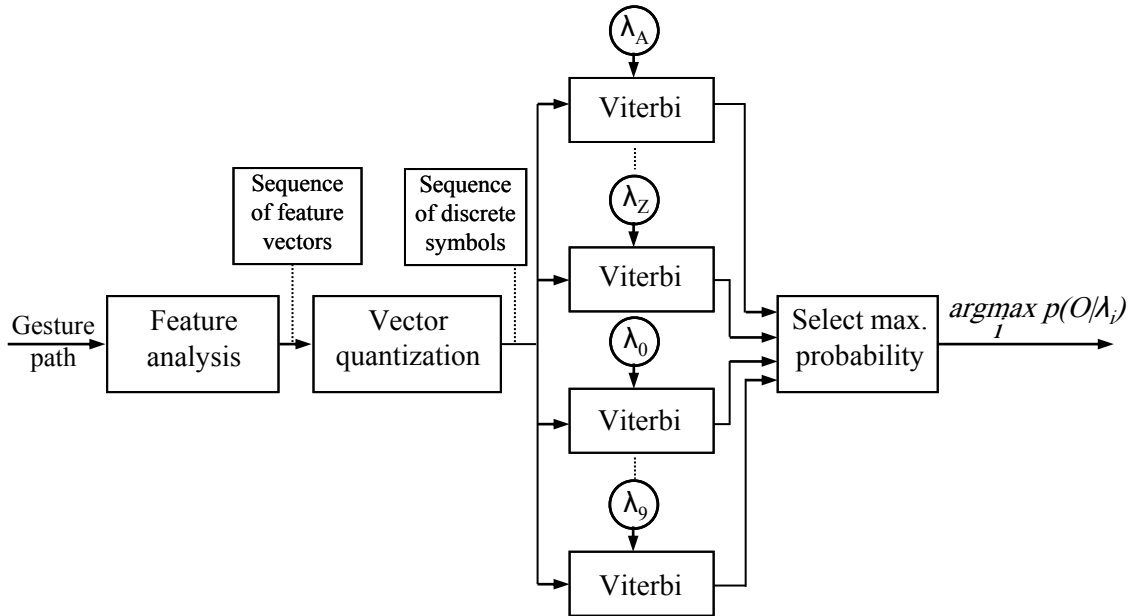


Figure 4.14: Block diagram of an isolated gestures by using HMMs (Viterbi) recognizer.

4.4.1.1 Model Size

Before the HMMs training starts, the size of HMMs must be decided. How many states do we need?

The number of states must be estimated by considering the complexity of the various patterns that HMMs will be used to distinguish. In other words, the number of segmented parts in the graphical pattern is taken into consideration when we represented it. When the number of training data samples is insufficient, the use of excessive state numbers causes the over-fitting problem³. In addition, the discrimination power of the HMMs is decreased when insufficient number of states is used because more than one segmented part of graphical pattern is modeled on one state. The number of states in our gesture recognition system is determined by mapping each straight-line segment into a single HMM state (Fig. 4.15). To represent various graphical patterns, we must look at the possible patterns and estimate how many distinguishable segments are contained in a pattern. It may be a good idea to use different numbers of states in the different HMMs, which used to represent separate classes of patterns. For example, to represent a graphical pattern 'L', only two states

³Over-fitting occurs when HMMs describe random error instead of the underlying relationship. Potential over-fitting problem does not only depend on the number of parameters and data, but also on the compatibility of model structure with the amount of model error and data shape. To avoid the problem of over-fitting, additional techniques (e.g. regularization, early stopping, cross-validation and etc.) are used when further training is not resulting in better generalization. For more details, the reader can refer to [129].

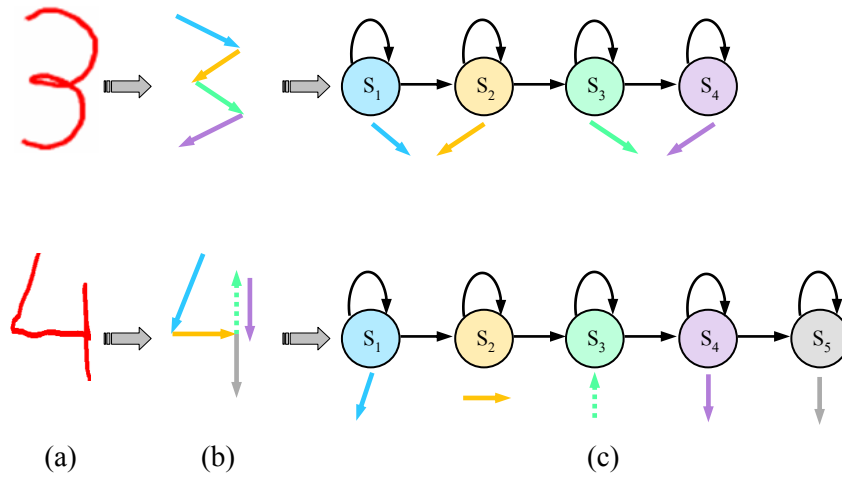


Figure 4.15: Straight-line segment for HMMs topologies (a) Gesture number from hand motion trajectory (b) Line segment of gesture number (c) LRB model with line segmented codewords.

are needed, whereas six states are required for a graphical pattern ‘E’, and four states for graphical pattern ‘3’.

4.4.1.2 Initializing a Left-Right Banded Model

Before starting the iterative Baum-Welch algorithm, the initial values of all parameters in the HMMs must be assigned. There is only one general requirement; the initial model must indicate, somehow, what we want to represent different model states. However, this requirement has different consequences, depending on the type of HMMs. In practice, the LRB model is considered because each state in Ergodic topology has many transitions than LR and LRB topologies, so, the structure data can be easily lost. On the other hand, LRB topology has no backward transition so, the state index either increases or remains the same as time increases. In addition, LRB topology is more restricted than LR topology and simple for training data, which can match the data to the model [93].

An intuitively observation is that, a good initialization for HMMs parameters (A, B, π) achieves better results. Matrix A is the first parameter, where it is determined using Eq. 4.26.

$$A = \begin{pmatrix} a_{11} & 1 - a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & 1 - a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \quad (4.26)$$

The diagonal elements a_{ii} of the transition matrix can be chosen to indicate approximately the average state durations d such that;

$$a_{ii} = 1 - \frac{1}{d} \quad (4.27)$$

and

$$d = \frac{T}{N} \quad (4.28)$$

where T is the length of gesture path and N represents the number of states.

This is sufficient for an automatic training procedure in which state 1 is intended to represent the first part of the training data, state 2 the next part, etc. Therefore, all output probability distributions for different states can be initialized with the same parameters for all states. Consequently, the first step in Baum-welch iteration uses the training data to calculate more correct output probability parameters for each state. Since HMMs states are discrete, all elements of matrix B are initialized with the same value for all different states (Eq. 4.30). Matrix B is an N -by- M observed symbols where b_{im} gives the probability of emitting symbol v_m in state i (Eq. 3.31).

$$b_{im} = \frac{1}{M} \quad (4.29)$$

where i, m run over the number of states and the number of discrete symbols, respectively.

$$B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1M} \\ b_{21} & b_{22} & \cdots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NM} \end{pmatrix} = \begin{pmatrix} \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} \\ \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} \end{pmatrix} \quad (4.30)$$

For each new time sample, the state can jump back by itself, or only to the nearest following state. Therefore, the initial probability vector π should be initialized as;

$$\pi = \left(1 \ 0 \ \cdots \ 0 \right)^T \quad (4.31)$$

It is to ensure that it begins from the first state.

4.4.1.3 Termination of HMMs Training

The Baum-Welch training algorithm is very efficient. Often a good model is reached already after 5-10 iterations. The trained model must be flexible enough to correctly represent a new test sequence that never occurred during training. The training step is repeated until the change of transition and emission matrix converges. The convergence is satisfied if the change is less than 0.001 (i.e. tolerance $\epsilon = 0.001$) as described in Eq. 4.32, or reaches to the maximum number of iterations (i.e. 500).

$$\sum_{i=1}^N \sum_{j=1}^N |\hat{a}_{ij} - a_{ij}| + \sum_{j=1}^N \sum_{m=1}^M |\hat{b}_{jm} - b_{jm}| < \epsilon \quad (4.32)$$

The main motivation behind using tolerance is to control the number steps required by the Baum-Welch algorithm in order to successfully execute its purpose. This algorithm is terminated if all of the following three quantities are less than the tolerance value. First, log-likelihood for a given observation sequence O is generated using the current estimated values of transition matrix A and observation matrix B . Second, change in the normalization of the transition matrix A . At the end, change in the normalization of the observation matrix B . Note that, increasing tolerance reduces the number of steps to execute the Baum-Welch algorithm before it was terminated. In fact, the maximum number of iterations controls the maximum number of steps to execute the algorithm. If the Baum-Welch algorithm executes 500 iterations before reaches to the specified tolerance value, the termination is occurred with a warning. When this occurs, the value of maximum number of iterations should be increased so that the algorithm reaches to the desired tolerance before termination.

It is usually very difficult to provide sufficient amounts of training data. Therefore, some observation may never occur in the limited set of training data, although we may know that they might have occurred with some small probability. If a discrete HMM is trained on a such data, the Baum-Welch will assign *zero* observation probability to some elements of the observation probability matrix. In such case, a very small non-zero value may be assigned and re-normalization of the row matrix is required. A similar problem can occur with the transition probability matrix. For a left-right banded HMM we have intentionally defined many elements of the transition probability matrix exactly *zero* values. These elements still have zero values after the Baum-Welch training, and should remain zero. Furthermore, the adjustment of HMMs parameters is important after performing the training operation.

4.4.2 Classification Using CRFs

CRFs use a single model of the joint probability of the label sequences (i.e. alphabets and numbers) given an observation sequence. Therefore, there are trade-off in the weights of occurrences of a feature value for each state [130]. The gesture recognition module matches the tested gesture against database of reference gestures, to classify which class it belongs to. Thereby, the hand gesture path is recognized corresponding to the maximal likelihood of all gestures (i.e. labels) accumulatively until it receives the gesture end signal. The maximal label of CRFs model is the gesture whose observation probability is the largest among all the gestures labels (Fig. 4.16).

4.4.2.1 Data Format of CRFs

CRFs and LDCRFs models are applied to unsegmented sequences while HCRFs should be apply to pre-segmented sequences (only one label per sequence). The data and the label files are encoded using Comma Separated Values (CSV) format

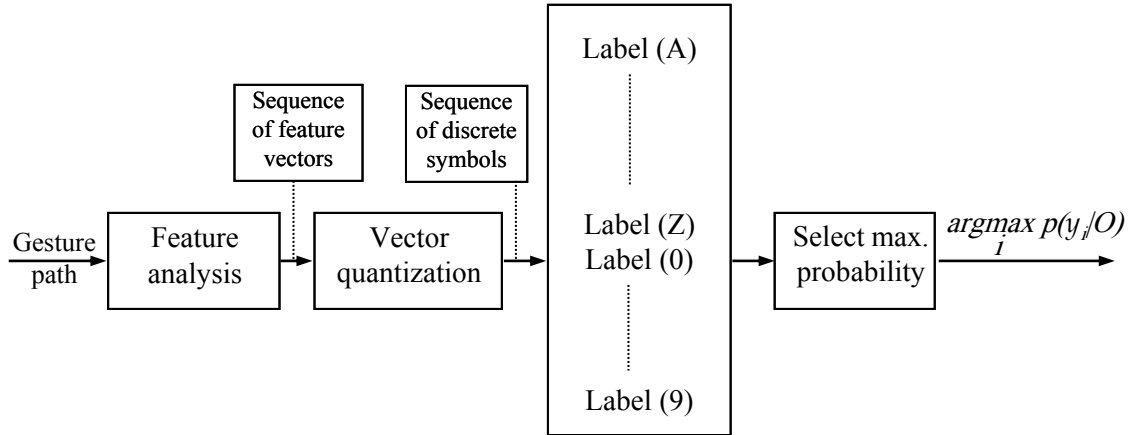


Figure 4.16: Block diagram of an isolated gesture using CRFs recognizer.

according to hcrf library⁴. Each file contains multiple matrices or vectors encoding the feature values (data files) or label values (label files). A data file contains multiple matrices, one for each sequence. For each matrix, the first line always contains two numbers: the number of rows and the number of columns. The number of rows for each matrix represents the number of features. All the matrices should have the same number of features. The number of columns for a specific matrix represent the number of time samples in the sequence.

Since HCRFs models have only one label associated to each sequence while CRFs and LDCRFs have one label associated to each time sample in the sequence, the HCRFs library supports two file format for labels. For HCRFs model, the label file contains one integer per line, representing the label for the specific sequence. For CRFs and LDCRFs models, the label file is encoded as a data file with matrix headers specifying the number of rows and columns but in this case the matrices always have one row. This row should have the same length as the corresponding sequence in the data file, with one label for each time sample.

4.4.2.2 Matching CRFs Model

The learning parameters of CRFs are based on the maximum entropy⁵, which is considered a good measure for the variational problems (e.g. a finite training data). In addition, maximum entropy has the ability to justify the probability distribution

⁴The CRFs formulation is implemented by extending the software of the library of Hidden-state Conditional Random Field [131]. This library implements three models: CRFs, HCRFs and LDCRFs with C++ and Matlab languages.

⁵The principle of maximum entropy: when one has partial data with regard to possible outcomes one should select the probabilities so as to maximize uncertainty with regard to the missing data, as shown by Jaynes [132]

from incomplete information. Likelihood maximization is employed using a gradient-based method in conjunction with BFGS optimization technique. BFGS technique solves nonlinear optimization problems which have limitations constraints in order to determine the best parameter values by using the smallest number of computational iterations. This technique uses first and second derivatives so that the gradient with zero represents a necessary condition for optimality. HCRFs and LDCRFs models have the same calculations in order to fully observe CRFs. For more details, the reader can refer to Section 3.4.2.1.

The models of HCRFs and LDCRFs are more restricted to the number of hidden states owned by each class label to make training and inferencing processes tractable. Furthermore, the training is done directly on the sub-structure of gesture sequences using intermediate hidden states. Each class label has a set of hidden states which significantly improve recognition performance, and powerful in simplifying models according to training and inferencing processes. The number of hidden states depends on the complexity of each hand gesture during training and inferencing processes. For instance, the number of hidden states per label is varied from 2 to 5 in our work because each label consists of segmented parts as shown in Fig. 4.15. However, each straight-line segment from gesture path for alphabets and numbers is mapped into a single hidden state. On a standard desktop, training process is more expensive for CRFs, HCRFs and LDCRFs than HMMs since the time which each model required it ranged from 20 minutes to several hours due to an observation window. Inference uses forward score of each sample to select the model with the highest likelihood. Additionally, inference is fast for all models (i.e. in seconds) for sequences of several frames. CRFs models with different input feature window size play a main role in system performance. A window size of zero means that the feature vector at the current frame is only used to construct the input feature. The window size of three means that the input feature vector at each frame consists of seven feature vectors: the current frame, the three preceding frames and the three future frames. Algorithm 2 summarizes the matching process of CRFs models for a given observation sequence.

4.5 Computational Complexity

For mean-shift algorithm, suppose K represents the average number of iterations per frame. It is being observed that the cost of weighted and non-weighted histograms are roughly equal because the values of their kernels are pre-calculated. In Eq. 4.12, the direction of hand's movement (i.e. centroid points) is determined by a division of two weighted sum of two terms. Mathematically, the mean cost of mean-shift algorithm for one scale is calculated as follows;

$$Mean_{Cost} = K \cdot (Cost_H + n_h Cost_D) \approx K n_h Cost_D \quad (4.33)$$

Input: An observation sequence O , T represents the length of O and the number of labels is L

Output: Probability of label sequence y given CRFs parameters: $p(y|O, \theta)$

```

i = 1, initialize Z
while i ≤ T do
  for j = 1 to L do
    for k = 1 to L do
      |  $M_i(y_j, y_k) = \exp(\sum_f \lambda_f t_f(y_j, y_k, O, i) + \sum_g \mu_g s_g(y_k, O, i))$ 
    end
  end
  Z = Z × Mi % Z is a normalization factor
  q* = Mi(yi-1, yi|O) % q* is the product of all matrices M
  i = i + 1
end
p(y|O,  $\theta$ ) =  $\frac{1}{Z} \times q^*$ 

```

Algorithm 2: Matching CRFs model

where $Cost_H$ refers to the cost of histogram and $Cost_D$ represents an additional cost for the division of two weighted sum of two terms.

Let the number of target pixels n_h has the same range for the number of histogram entries u (i.e. in our work, $u = 1, \dots, 16$). The actual target histogram is determined and updated by sliding $\sqrt{n_h}$ vertical steps and $\sqrt{n_h}$ horizontal steps. Thus, the effort E is computed by;

$$E = Cost_H + 2n_h \sqrt{n_h} Cost_{add} \quad (4.34)$$

where $Cost_{add}$ represents an additional cost. Then, the total effort for target localization is almost equal;

$$Cost_E = Cost_H + 2n_h \sqrt{n_h} Cost_{add} + (u + 2n_h \sqrt{n_h}) \approx 2n_h \sqrt{n_h} Cost_D \quad (4.35)$$

The ratio between Eq. 4.33 and Eq. 4.35 is equal to $2 * \sqrt{n_h} / K$. In our work, the target represents 16×16 pixels (i.e. $\sqrt{n_h} = 16$) and the mean number of iterations per frame is $K = 1.61$ (Fig. A.6). So, the optimization process for mean-shift procedure decreases the time of computation $2 * 16 / 1.6 \approx 20$ times. It is noted that the computational time should be multiplied by three in case of scale adaptations for hand target.

The time complexity of the CRFs matching algorithms presented in this chapter is proportional to the number of cells which are visited by dynamic programming method. CRFs takes $O(TL^2)$ where L is the number of labels (i.e. in our case the alphabets and numbers) and T is the number of input feature vectors at every time instance. The space complexity of the matching algorithm is similar to the time complexity if the algorithm is running in offline and online modes.

4.6 Discussion and Conclusion

In this chapter, the proposed system for isolated gestures (e.g. alphabets and numbers) has been described from image acquisition to classification phase. In the first step, one of the main contributions of this work was to exploit depth image sequences. The obtained depth information from stereo camera system defines the ROI instead of processing whole image which consequently reduces the cost of ROI searching and increases the processing speed. Furthermore, the depth information has been used to increase the accuracy of objects segmentation as well as identifying the objects under occlusion.

Improvements and extensions have been carried out for gesture system in the second step. Precisely, a robust method for hand tracking in complex environment using mean-shift algorithm in conjunction with depth map was proposed. Mean-shift analysis used the gradient of Bhattacharyya coefficient as a similarity function to derive the candidate of the hand which is most similar to a given hand target model. This structure extracts a set of hand postures to track the hand motion, and achieves accurate and robust hand tracking with a Bumblebee stereo camera as an input device. The input images are unstable due to the changes in lighting conditions, background color and hand shaking during movement. So, it causes frequent, sharp changes of the centroid or fingertip points. To alleviate these changes, the spatio-temporal trajectories are smoothed as the mean values of a specified point with its neighbors points. In the third step, the features of location, orientation and velocity (which are obtained from spatio-temporal hand gesture path) with respect to Cartesian and Polar systems are combined and analyzed. This analysis determines the degree of effectiveness of such combination on the recognition rates.

Classification is the final step in our proposed system. Classification of the symbols in gesture recognition assigns them to a respective class. Throughout this stage, the isolated gestures were handled according to two different classification techniques: a generative model such as HMMs and discriminative models like CRFs, HCRFs and LDCRFs. In addition, HMMs using Ergodic, Left-Right and Left-Right Banded topologies with different number of states ranging from 3 to 10 have been analyzed and studied in terms of their impact on gesture recognition. Furthermore, this research contributes on the decision of which HMMs topology and classification technique is the optimal in term of results. The next chapter demonstrates the experimental results and the analysis of isolated hand gestures.

Chapter 5

Isolated Gesture Recognition Test

5.1 Data Set

The alphabets and numbers are classified using HMMs, CRFs, HCRFs and LDCRFs by the motion trajectory of single hand. A database is developed containing 2160 video samples for gesture symbols taken from three subjects on a set of 26 alphabets and 10 numbers. In other words, each isolated gesture is based on 60 video sequences where 42 video samples for training and 18 video samples for testing (In total, our database contains 1512 video samples for training and 648 video samples for testing). The sample test data is entirely different from the training data and is tested on *Intel(R) Core(TM)2 Duo CPU 2.2GHz PC with 4 GB of RAM*. The input images are captured by Bumblebee stereo camera system which has 6 mm focal length at 15FPS with 240×320 pixels image resolution, and Matlab implementation. Bumblebee camera is used for acquisition of 2D images along with depth map. Therefore the databases are captured in IESK lab¹, Otto-von-Guericke-University Magdeburg, Germany (Fig. 5.1).

5.2 Experimental Discussion

A method for detection and segmentation of the hands in stereo color images is developed with complex background where the hand segmentation and tracking take place using depth map, color information, GMMs and mean-shift algorithm. Firstly, segmentation of skin colored regions becomes robust if chrominance components are used in analysis. Therefore, YC_bC_r color space is used in our system where Y channel represents brightness and (C_b, C_r) channels refer to chrominance. The luminance channel Y is ignored to reduce the effect of brightness variation and use only the chrominance channels, which fully represent the color information. A large database of skin and non-skin pixels is used to train the Gaussian model. GMMs technique

¹<http://www.iesk.ovgu.de/>



Figure 5.1: IESK lab.

begins with modeling of skin pixels using skin database where a variant of k -means clustering algorithm performs the training model to determine the initial configuration of GMMs parameters. Additionally, blob analysis is used to derive the hand boundary area, centroid point and bounding box.

Secondly, after localization of the hand's target from segmentation step, its color histogram is considered with Epanechnikov kernel. This kernel assigns smaller weights to increase the robustness of the density estimation. To find the optimal match of hand target in sequential frames, the Bhattacharyya coefficient is used to measure the similarity by maximizing Bayes error which arises from the comparison of hand target and candidate. The computed mean depth value from the previous frame for hand region is taken into consideration. The depth information is used to define the region of interest instead of processing whole image to increase the processing speed as well as it resolves the complex background. Mean-shift procedure is recursively defined and performs the optimization to compute the mean-shift vector. Thereby, the hand gesture path is obtained by taking the correspondences of detected hand among the successive frames. Combined features of location, orientation and velocity with respect to Cartesian and Polar systems are used to increase the recognition rate. After that, k -means clustering is employed for HMMs, CRFs, HCRFs and LDCRFs codewords.

5.3 Experimental Results and Analysis

Our proposed system is capable for real-time implementation and showed good results to recognize isolated alphabets and numbers from stereo color image sequences. Our experiments are carried out on isolated gestures according to two different classification techniques: generative model such as HMMs and discriminative models like CRFs, HCRFs and LDCRFs. The following sections discuss the analysis of HMMs and CRFs results in details.

5.3.1 HMMs

In our experimental results, each isolated gesture was based on 60 video sequences in which 42 video samples for training by Baum-Welch algorithm and 18 video samples for testing (i.e. in total, our database contains 1512 video samples for training and 648 video samples for testing). The gesture recognition module matches the hand gesture path against the database of reference gestures to classify in the class it belongs to. The higher priority has been computed by Viterbi algorithm to recognize the alphabets and numbers frame by frame.

There is no doubt that selecting good features to recognize the hand gesture path plays a significant role in system performance. In addition, the selection of the best HMMs topology plays an important role in the classification process and is presented in the following subsections.

5.3.1.1 Feature Extraction Analysis

The main contribution of this section is to examine the capabilities of combined features of location, orientation and velocity for gesture recognition. These features are obtained from spatio-temporal hand gesture path. The importance of these features are tested according to Cartesian and Polar coordinate systems. Furthermore, experiments with varying features are performed to decide the best features in term of results. The observation sequence for Left-right banded model is quantified either by using the normalization in case of separated features or by using the k -means clustering algorithm in case of combined features. For more details, the reader can refer to Section 4.3.

According to the separated features in Fig. 5.2 (a) & (b), the orientation features ($\theta_1, \theta_2, \theta_3$) are better in recognition rate than the recognition rate of location features (Lc, Lsc) or velocity feature (V). This in turn leads to the orientation feature ($\theta_1 = 93.06\%$) to be the most effective among the three basic features (i.e. location, orientation and velocity). Furthermore, the velocity feature with 57.25% recognition rate represents a lower discrimination power than the orientation features because there is a quite bit of variability (i.e. varying speed during gesture generation) in the same gesture even for the same person. Also, Lsc feature result has the lowest recognition rate of 32.72%. In general, the testing results from the union of features show that the combined features in Cartesian system yield a higher recognition ratio than the combined features in Polar system (Table 5.1). Additionally, the (Lc, Lsc, V), ($\theta_1, \theta_2, \theta_3, V$) and ($Lc, Lsc, \theta_1, \theta_2, \theta_3, V$) features which contain the velocity information provide higher recognition rate than the use of velocity feature alone (Fig. 5.2(c)). But lower recognition results are observed in case of Polar coordinate (Fig. 5.3(a)).

Fig. 5.2 shows the results of the experiments which have been performed to determine the optimal feature code numbers. Here, k -means is a coding method for converting location, orientation and velocity values to feature code (i.e. codeword)

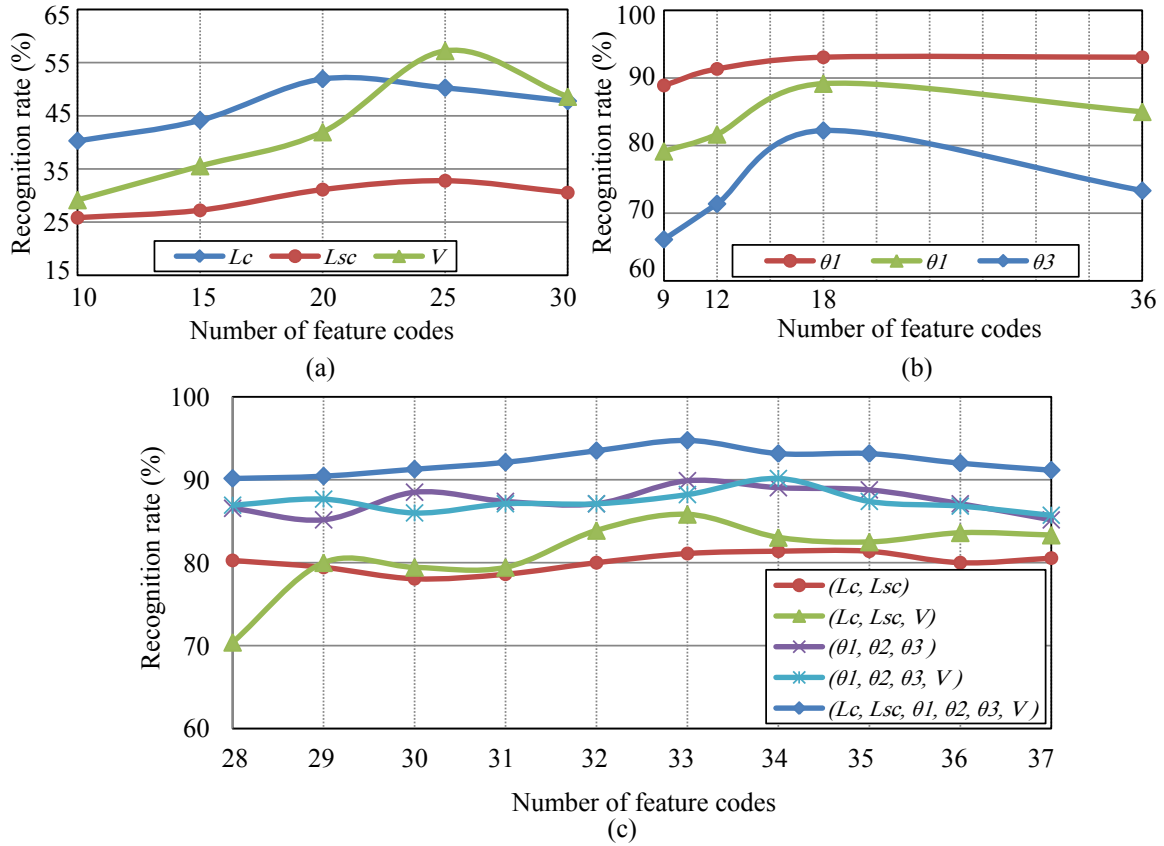


Figure 5.2: The number of feature codes represents either the number of clusters in case of combined features or the number of normalized codewords in case of separated features. (a) The recognition of locations and velocity features according to different number of codewords (10, 15, 20, 25, 30). (b) Results for three different orientations with varying feature codewords number (9, 12, 18, 36). (c) Recognition rate of different combined features in Cartesian system with different codewords number ranging from 28 to 37.

which represents an element of a standardized code (for instance, cluster numbers in our application). The optimal number of feature code is 33 for the combined features $(L_c, L_{sc}, \theta_1, \theta_2, \theta_3, V)$. Fig. 5.2(b) shows the system output for isolated gesture number ‘3’ in addition to the solved overlapping problem between hand and face by using depth map. The cluster trajectories for gestures numbers (0-9) are depicted in Fig. A.3, Fig. A.4 and Fig. A.5 (Appendix A).

In short, the effectiveness of these features yields reasonable recognition rates. The proposed system has shown good performance when applied on several video samples containing confusing situations such as partial occlusion and overlapping. The results show that the proposed system successfully recognizes hand gestures with 94.75% recognition rate. From table 5.1, the recognition ratio of isolated gestures achieves

Table 5.1: Results of isolated gestures according to different features extraction in Cartesian and Polar systems with optimal feature code number.

Feature type	Feature space	Number of feature code	Training data	Isolated gestures results		
				Testing data	Correct data	Recognition (%)
Separated in Cartesian coordinates	L_c	20	1512	648	337	52.01
	L_{sc}	25	1512	648	212	32.72
	V	25	1512	648	371	57.25
Union in Cartesian coordinates	θ_1	18 ; 36	1512	648	603	93.06
	θ_2	18	1512	648	578	89.20
	θ_3	18	1512	648	533	82.25
Union in Cartesian coordinates	(L_c, L_{sc})	35	1512	648	527	81.33
	(L_c, L_{sc}, V)	33	1512	648	556	85.80
	$(\theta_1, \theta_2, \theta_3)$	33	1512	648	608	93.83
Union in Polar coordinates	$(\theta_1, \theta_2, \theta_3, V)$	34	1512	648	610	94.14
	$(L_c, L_{sc}, \theta_1, \theta_2, \theta_3, V)$	33	1512	648	614	94.75
	(ρ_c, φ_c)	28	1512	648	607	93.67
Union in Polar coordinates	$(\rho_{sc}, \varphi_{sc})$	33	1512	648	599	92.44
	(ρ_c, φ_c, V)	31	1512	648	604	93.21
	$(\rho_{sc}, \varphi_{sc}, V)$	30	1512	648	586	90.43
	$(\rho_c, \varphi_c, \rho_{sc}, \varphi_{sc}, V)$	29 ; 30	1512	648	591	91.20

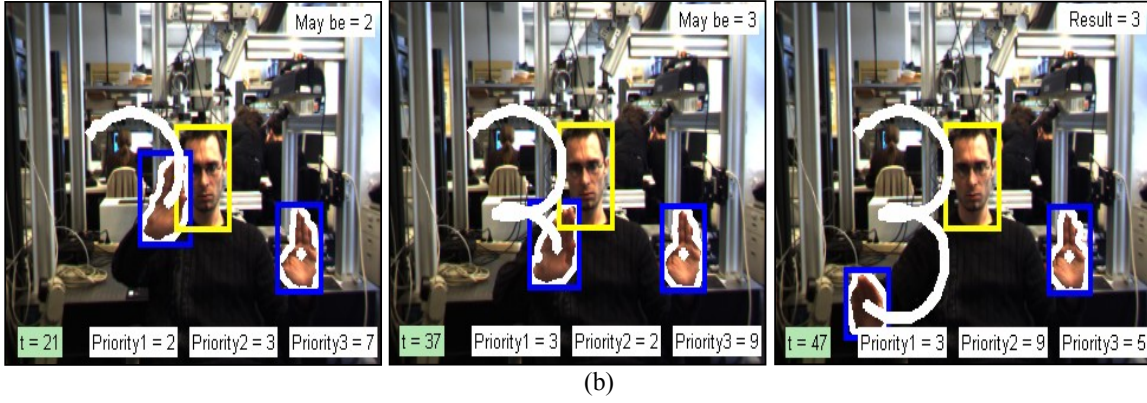
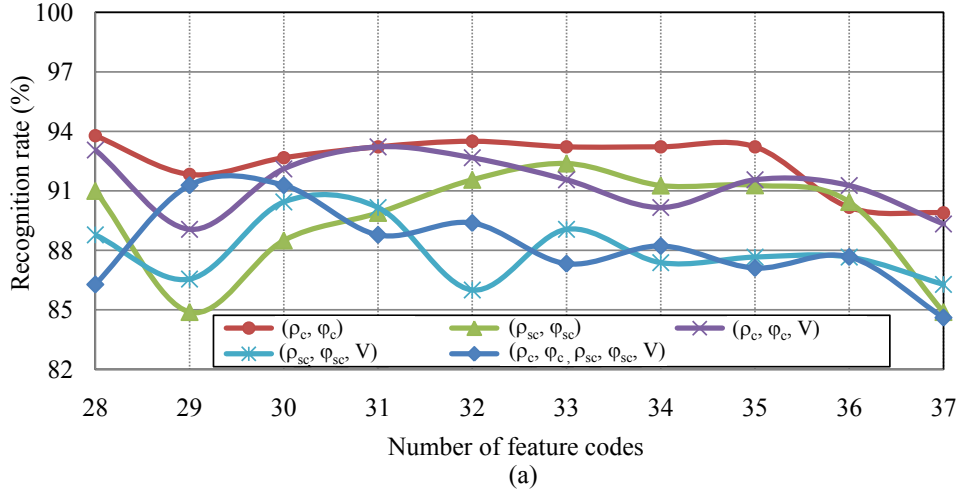


Figure 5.3: (a) Recognition rate according to combined features in Polar system with different feature codewords number ranging from 28 to 37. (b) The highest priority at $t = 21$ is gesture number ‘2’ and at $t = 47$ the final result is gesture number ‘3’.

best results using $(Lc, Lsc, \theta_1, \theta_2, \theta_3, V)$ features. The recognition ratio is the number of correctly recognized gestures to the number of tested gestures (Eq. 5.1).

$$\text{Recognition ratio} = \frac{\# \text{ recognized gestures}}{\# \text{ test gestures}} \times 100\% \quad (5.1)$$

5.3.1.2 Analysis Results of HMMs Topologies

In this thesis, the focus is to design HMMs topologies with different number of states to decide the best topology in term of results for isolated gestures system. HMMs using Ergodic, Left-Right (LR) and Left-Right Banded (LRB) topologies are applied on a discrete vector feature which is extracted from stereo color image sequences. These topologies are considered with different number of states ranging from 3 to 10. The number of states in our gesture recognition system is based on the complexity of

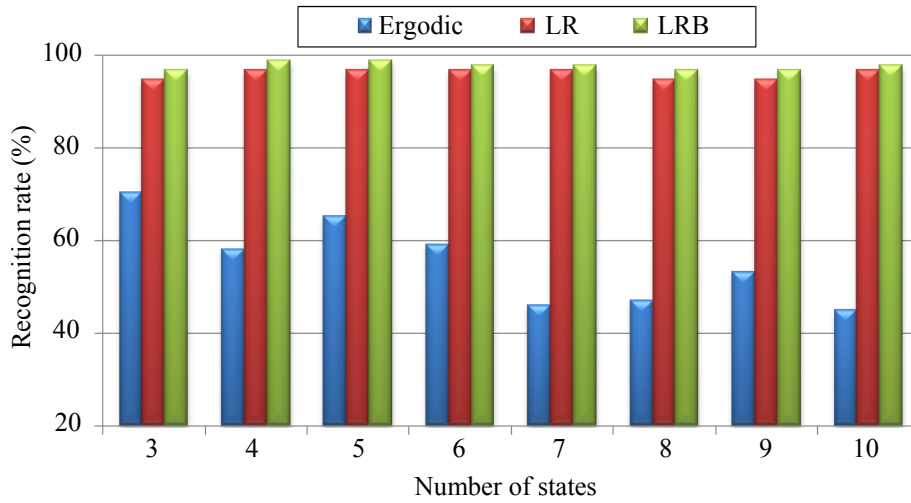


Figure 5.4: Isolated gesture recognition results for HMMs topologies with number of states ranging from 3 to 10.

each gesture number and is determined by mapping each straight-line segment into a single HMMs state.

The number of states is an important parameter for two reasons. First, when the number of training data samples is insufficient, the use of excessive state numbers cause the over-fitting problem. Second, the discrimination power of HMMs is decreased when using insufficient number of states because more than one segmented part of graphical pattern is modeled on one state.

In practice, to ensure that all states are used, the LRB model with 5 states is employed for gesture recognition system. Since each state in Ergodic topology has many transitions rather than LR and LRB topologies, the structure data can easily be lost. On the other hand, LRB topology has no backward transition where the state index either increases or stays the same as time increases. In addition, LRB topology is more restricted than LR topology and simple for training the data which will be able to match the data to the model. Also, the gesture paths ‘4’ and ‘5’ contain the largest number of segmented part and to ensure that all these parts are used, the use of 5 states are considered. For more details the reader can refer to [91,95], Section 4.4.1.1 and Fig. B.2 & Fig. B.3 (Appendix B).

In this experiment, each isolated gesture number (0-9) is based on 60 video sequences in which 42 video samples for training and 18 video samples for testing. In other words, our database contains 420 video sequences for training and 180 video sequences for testing the isolated gestures. The HMMs topologies are trained by BW algorithm and tested using Viterbi algorithm. From Fig. 5.4, the LRB presents the best performance where the average ratio of LRB topology from 3 to 10 states is 97.78%. Also, LR and LRB topologies with 4 states achieved the best recognition.

In addition, LRB topology is always better than LR and Ergodic topologies. In Fig. 5.4, there is no large gap between LRB and LR in terms of results but the results of Ergodic topology was poor when compared to LRB and LR topologies. In general, LRB topology with number of states equal to 5 is the best in terms of their impact on gesture recognition empirically, which in turn confirms the existing theoretical discourse in Section 4.4.1.1.

5.3.2 CRFs, HCRFs and LDCRFs

In CRFs experimental results, each alphabet and number were based on 60 videos, which contain 42 for training and 18 for testing. A CRFs, HCRFs and LDCRFs were constructed using combined features of location, orientation and velocity as described in Section 4.3. To handle isolated gesture, CRFs, HCRFs and LDCRFs with different number of window sizes (W) ranging from 0 to 7 are applied and tested to decide the best in term of recognition results.

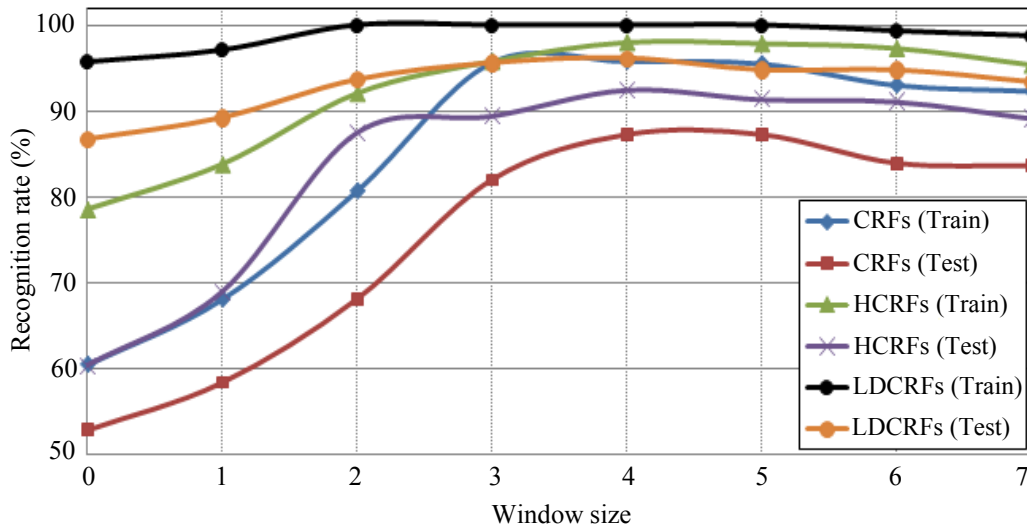


Figure 5.5: Recognition accuracy with different window sizes (0-7) for CRFs, HCRFs and LDCRFs on training and testing data.

A window size of zero means that the feature vector at the current frame is only used to construct the input feature while the window size of three means that the input feature vector at each frame consists of seven feature vectors which contain the current frame, three preceding frames and three future frames. In our application, the size of window is based on the complexity of each gesture as described in previous section. So, multiple experiments have been conducted with a variety of window size from 0 to 7 on the proposed system to empirically conclude the optimal outcome of the system. Fig. 5.5 shows the recognition rate of CRFs, HCRFs and LDCRFs according

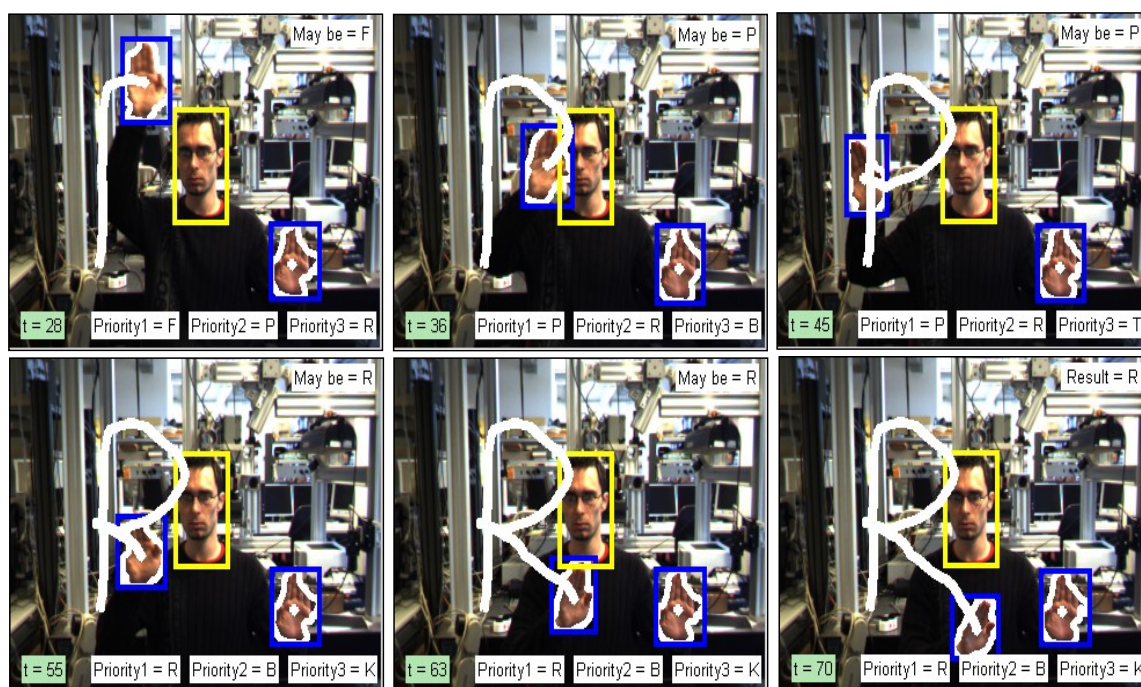
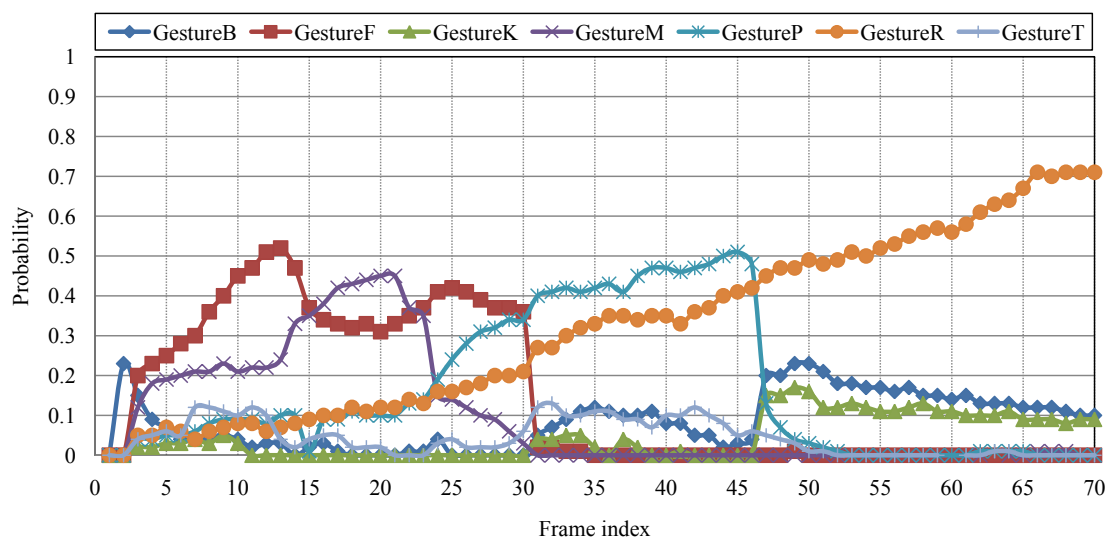


Figure 5.6: Temporal evolution of the seven higher probabilities of the gestures ‘B’, ‘F’, ‘K’, ‘M’, ‘P’, ‘R’ and ‘T’ using CRFs. In the image sequences, the high priority is alphabet ‘F’ at $t = 28$, at $t = 45$ the high priority is alphabet ‘P’ and at $t = 70$ the result is ‘R’. The hand motion trajectory is generated by connecting the centroid point of hand region.

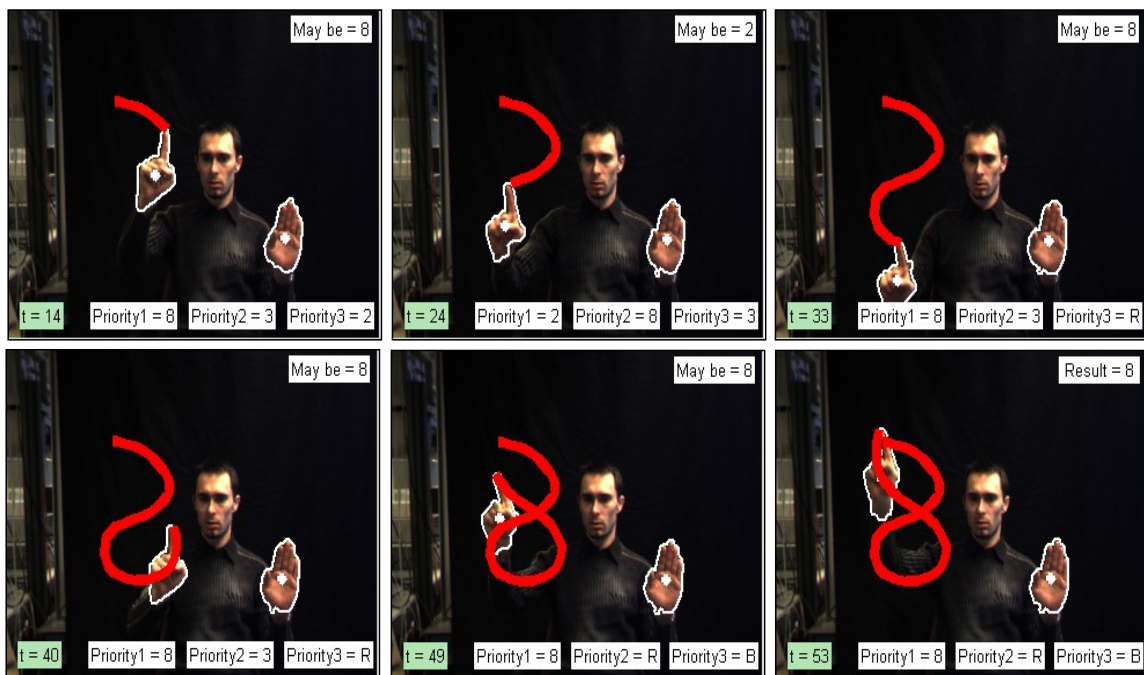
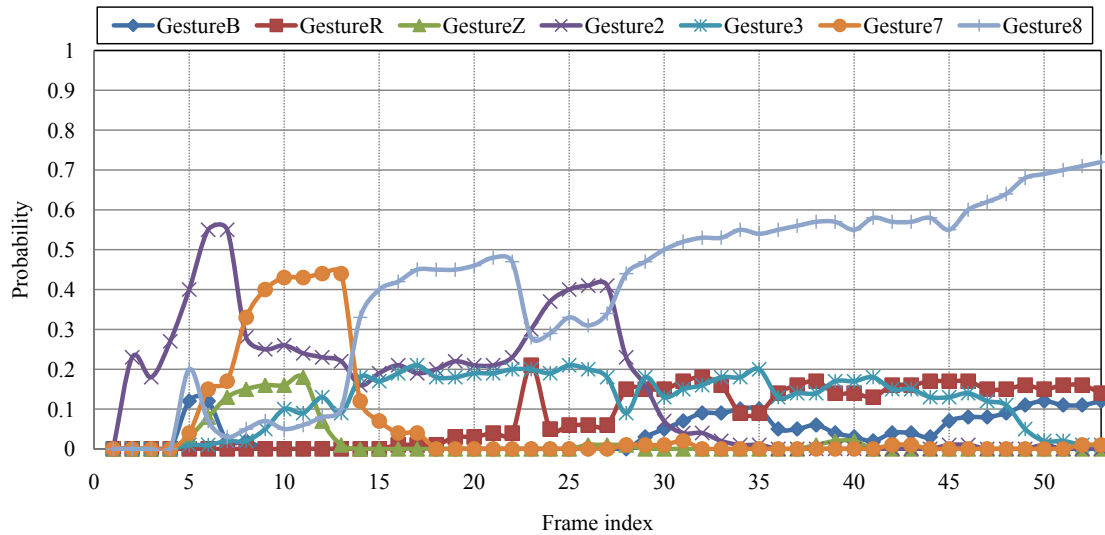


Figure 5.7: Temporal evolution of the seven higher probabilities of the gestures ‘B’, ‘R’, ‘Z’, ‘2’, ‘3’, ‘7’ and ‘8’ using HCRFs. In the image sequences, the high priority is number ‘2’ at $t = 24$, at $t = 40$ the high priority is number ‘8’ and at $t = 53$ the result is ‘8’. The hand motion trajectory is generated by connecting the fingertip points of the region of interest.

to different window sizes for training and testing data. The recognition of hand gesture path using LDCRFs is higher than CRFs and HCRFs. In addition, the yield of training data is higher than testing data in the proposed system. Furthermore, the gesture recognition rate is initially improved as the window size increases but degrades as the window size further increases. Therefore, the proposed system achieves better recognition at window size equal to 4 where it was automatically recognize tested gestures with 87.19%, 92.44%, 96.14% for CRFs, HCRFs and LDCRFs, respectively. Fig. 5.6 and Fig. 5.7 show the results of gesture paths ‘R’ and ‘8’ according to the seven higher probabilities using CRFs and HCRFs models, respectively. In addition, Fig. B.4 in Appendix B illustrates the result of gesture path ‘6’ using LDCRFs model.

5.3.3 Generative Model versus Discriminative Models

The difference between HMMs and CRFs is that HMMs are the generative models and define a joint probability distribution to solve a conditional problem, thus focusing on modeling the observation to compute the conditional probability. Moreover, one HMM is constructed per label (i.e each alphabet or number) where HMMs assume that all the observations are independent. Whereas CRFs are undirected graphical models and are developed for labeling sequential data. The key features of CRFs than HMMs are represented in their conditional nature and the dependencies assumptions of their computations to ensure tractable inference. In addition, CRFs overcome the weakness of directed graphical models which suffer from the bias problem as in MEMMs [26]. Furthermore, CRFs combine the strength of MEMMs and HMMs where they have all the characteristics of the directed graphical models as in HMMs. In addition, each label in CRFs is employed as exponential model as in MEMMs to conditional probabilities of the next label for a given current label. CRFs use a single model for all alphabets and numbers.

HCRFs models are the extension of CRFs which incorporate hidden state variables to deal well with gesture sub-structure [39, 102]. The main advantage of HCRFs is to automatically model the local interconnection between labels with hidden variables. On the contrary, they can not model the dynamics among states. LDCRFs models have the ability to overcome the main weaknesses of HCRFs models. In addition, LDCRFs models combine the advantages of CRFs and HCRFs where they learn extrinsic dynamics by modeling the class labels as well as they learn the intrinsic sub-structure of gesture sequence using intermediate hidden states. LDCRFs models are naturally used to recognize the un-segmented sequences because they contain a class label per observation as described in Section 3.4.2.3. Furthermore, LDCRFs models efficiently infer the gesture sequences during the training and testing processes.

Several experiments were run to compare between generative model like HMMs and discriminative models like CRFs, HCRFs, LDCRFs [93, 130]. In HMMs, we use a Left-Right Banded model based on Gaussian emission probabilities which have

Table 5.2: Results of gestures recognition at $W = 0$

Model type	Data set		Recognition result (%)		
	Training	Testing	Training	Testing	Overall
CRFs	1512	648	60.34	52.78	56.56
HCRFs	1512	648	78.55	60.34	69.45
LDCRFs	1512	648	95.68	86.73	91.21
HMMs	1512	648	99.07	94.75	96.91

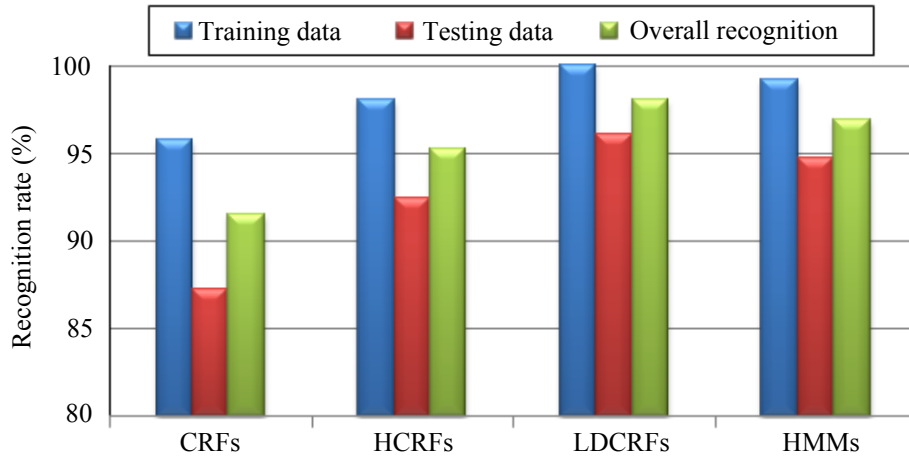


Figure 5.8: Results of gestures recognition using CRFs, HCRFs, LDCRFs versus HMMs at window size = 4.

a full covariance matrix for each state. The HMMs parameters (i.e. the emission probability and the state transition matrix) are learned from the same training data used by CRFs. HMMs are trained by BW algorithm while CRFs are trained using gradient ascent with the BFGS optimization technique [26] with 300 iteration to converge. On a standard desktop PC, training process is more expensive for CRFs, HCRFs and LDCRFs than HMMs since the required time to model ranges from 20 minutes to several hours and is based on observation window. On the contrary, the inference (i.e. recognition) process is less costly and very fast for all models with sequences of several frames (e.g. more than 80 frames in a sequence). The type of observed gesture is decided with HMMs by Viterbi algorithm, frame by frame. As shown in Table 5.2, the overall recognition rate (the average of the training and the testing of recognition result) of HCRFs at window size equal to 0 is higher than CRFs. Also, in that case, the overall recognition rate achieved by HMMs is 96.91%.

Furthermore, HMMs is the best in terms of results than CRFs, HCRFs and LDCRFs at $W = 0$. Whereas at window size equal to 4, LDCRFs recognition rate is higher than HMMs according to the training and the testing data (Fig. 5.8). Our results show that the overall recognition rates are 91.51%, 95.22%, 96.91% and 97.99% for CRFs, HCRFs, HMMs and LDCRFs, respectively. The high recognition rate achieved by the proposed system is due to the following reasons; 1) As a benefit of depth information, a high segmentation accuracy of the hand is achieved. 2) A set of feature candidates that optimally discriminate among the input patterns is elected. 3) A carefully experimental based selection of initialization parameters for training process. 4) HMMs, CRFs, HCRFs and LDCRFs classification techniques have the ability to efficiently alleviate spatio-temporal variabilities.

5.4 Discussion and Conclusion

In this chapter, experiments were carried out on isolated gestures according to two different classification techniques: a generative model such as HMMs and discriminative models like CRFs, HCRFs and LDCRFs.

For HMMs, the main contribution was to examine the capabilities of the combined features of location, orientation and velocity with respect to Cartesian and Polar coordinates. It has been shown that the effectiveness of these features yields reasonable recognition rates. The velocity and location features showed a lower discrimination power than orientation feature. Furthermore, the proposed system successfully recognizes isolated hand gestures with 94.75% recognition rate using $(Lc, Lsc, \theta_1, \theta_2, \theta_3, V)$ feature. Another contribution was to handle HMMs topologies with different states to decide best topology in terms of their impact on gesture recognition. It is concluded that there is no large gap between Left-right Banded (LRB) model and Left-right (LR) model in the recognition rates. On the contrary, the results of Ergodic topology was poor when compared to LRB and LR topologies. In general, LRB topology with 5 states was the best in term of results.

For discriminative models, CRFs, HCRFs and LDCRFs with different number of window sizes ranging from 0 to 7 were applied and tested to decide the best among them. It is concluded that the optimal size of window is equal to 4 empirically, where the proposed system automatically recognizes tested gestures with 87.19%, 92.44%, 96.14% for CRFs, HCRFs and LDCRFs respectively. In contrast to generative and discriminative models, HMMs was the best in terms of results than CRFs, HCRFs and LDCRFs at window size = 0. Whereas at window size equal to 4, LDCRFs recognition results were higher than HMMs according to the training and the testing data. Our results showed that, the overall recognition rates were 91.51%, 95.22%, 96.91% and 97.99% for CRFs, HCRFs, HMMs and LDCRFs, respectively. It is noted that the proposed system achieves high recognition rate due to a high segmentation accuracy of hand through the use of depth information. In addition, a good election

for the set of feature candidates that optimally discriminate among input patterns. Also, a careful experimental based selection is required for initialization parameters of training process. Above all, HMMs, CRFs, HCRFs and LDCRFs classification techniques have the ability to efficiently alleviate spatio-temporal variabilities. The next chapter will explore hand gesture spotting (i.e. extracting meaningful gestures from continuous hand motion) and recognition using HMMs and CRFs.

Chapter 6

Gesture Spotting and Recognition

While automatic hand gesture recognition technology has been exists which applied to real-world applications, there are still several problems which need to be solved for wider applications of HCI. One of such problems, in hand gesture recognition is to extract (spot) meaningful gestures from the continuous sequence of hand motions. Another problem is due to the variability (i.e. varies in shape, trajectory and duration) in the same gesture even for the same person. The goal of gesture spotting and interpretation is to make the human-machine communication close to human-human interaction. Gesture can be divided into two types; communicative gesture (key/meaningful gesture) and noncommunicative gesture (garbage gesture or transition gesture) [6, 64]. In other words, a natural gesture includes three phases: pre-, key- and post-gesture as in Fig. 6.1. The key gesture is defined as a part of hand trajectory that carries explicitly meaning for human. Whereas, pre- and post-gestures represent unintentional movements which are used to connect key gestures. Fig. 6.1 illustrates how a gesture path can be implemented with different phases and spotted in spatio-temporal space.

Previous approaches mostly use backward spotting technique to first detect the end point of gesture by comparing the probability of maximal gesture models and non-gesture model [15, 16, 32, 34]. Secondly, they track back to discover the start point of gesture through their optimal path and recognize the segmented gesture. Thus, there is a time delay between key gesture spotting and recognition and this time delay is unacceptable for on-line applications. In addition, there is an inadequate research to address the problems on non-gesture patterns (i.e. pre- and post-gestures) for gesture spotting because the number of non-gesture patterns is infinity, which in turn lead to the difficulty of modeling non-gesture patterns.

The main contribution of this chapter is to propose a forward gesture spotting scheme which handles hand gesture spotting and recognition of numbers¹ (0-9) simultaneously. This scheme uses a stochastic method for designing a non-gesture model by

¹An application of gesture-based interaction with numbers is implemented to demonstrate the coaction of suggested components and the effectiveness of gesture spotting and recognition system.

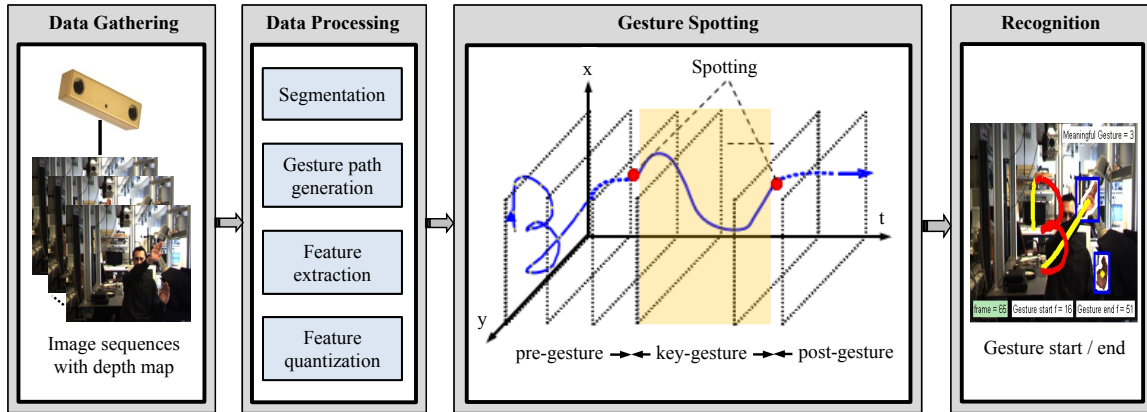


Figure 6.1: Concept of the hand gesture spotting and recognition system.

HMMs versus CRFs models with no training data. Furthermore, this scheme solves the issues of time delay between the spotting and the recognition task. The following sections describe how HMMs and CRFs are used for hand gesture spotting and recognition. In addition, how to model gesture patterns discriminately and how to model non-gesture patterns effectively without training data for non-gesture patterns.

6.1 Spotting with HMMs

HMMs are capable of modeling spatio-temporal time series of gestures effectively and can handle non-gesture patterns (garbage model or filler model) than NN and DTW. To spot key gestures accurately, a non-gesture model is proposed. The non-gesture model provides a confidence measure based on the calculated likelihood of gesture models which is used as an adaptive threshold to find the start and the end points of key gestures which are embedded in the input video sequences. The performance of non-gesture model is improved using relative entropy measure to alleviate the problem of increasing number of states [105]. The following subsections explain the stages of spotting and designing a non-gesture model from gestures models (Fig. 6.2).

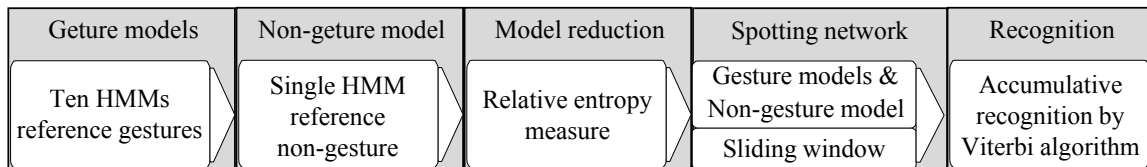


Figure 6.2: Road map of gesture spotting and recognition using HMMs.

6.1.1 Gesture Model

For each reference gesture, each HMM state represents its local segmental part. However, the transition among states represent the sequential order structure in a gesture

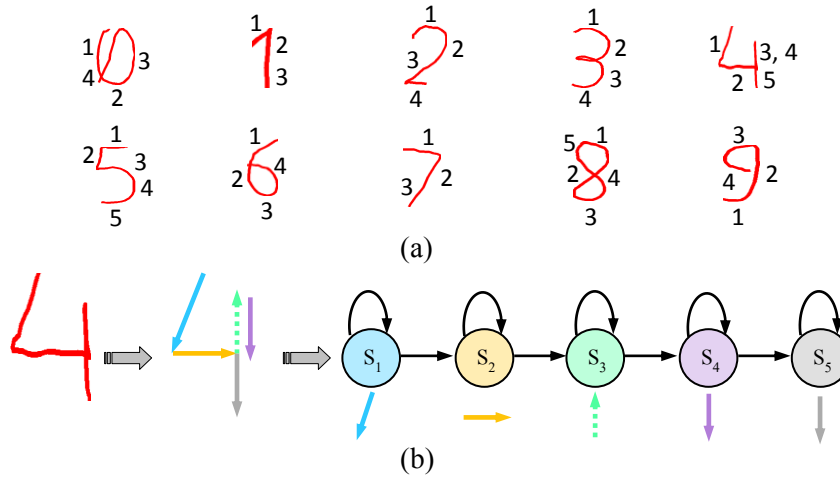


Figure 6.3: The hand gesture paths and straight-line segmentation. (a) The gesture paths from hand motion trajectory for numbers (0-9) with its segmented parts. (b) The LRB topology with segmented line for a gesture path ‘4’.

path. The number of HMMs states is an important parameter for each reference gesture. When the number of training data samples is insufficient, the use of excessive state numbers cause the over-fitting problem. In addition, the discrimination power of HMMs is decreased in case of using insufficient number of states, because more than one segmented part of graphical pattern is modeled on one state. Moreover, the number of states in our gesture spotting system is based on the complexity of each gesture number and is determined by mapping each straight-line segment into a single HMM state (Fig. 6.3).

In practice, the LRB model is considered for the following reasons. Since each state in Ergodic topology has many transitions than LR and LRB topologies, the structure data can be lost easily. On the other hand, LRB topology has no backward transition where the state index either increases or stays the same as time increases. In addition, LRB topology is more restricted than LR topology and simple for training data and is able to match the data with the model. Therefore, Baum-Welch algorithm plays a significant role in our system, where it is used to do a full training for the initialized HMMs parameters $\lambda = (\pi, A, B)$. For more details, the reader can refer to Section 4.4.1.2.

6.1.2 Non-gesture Model

It is not easy to obtain the set of non-gesture patterns because there are infinite varieties of meaningless motion. So, all other patterns other than reference patterns are modeled by a single HMM called a non-gesture model (garbage model) [15, 133, 134, 135]. A non-gesture model represents any motion trajectory or any part of it other

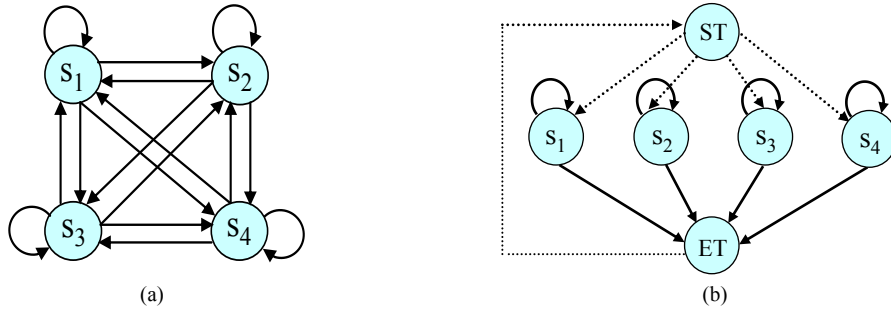


Figure 6.4: (a) Ergodic topology (b) Simplified ergodic with two dummy states and fewer transitions.

than gesture model. For the correct gesture spotting, the likelihood of a gesture model for a given pattern (i.e. previously mentioned) should be distinct enough. Although the HMMs recognizer selects a model with the best likelihood, but we cannot ensure that the pattern is really similar to the reference gesture model unless the likelihood value is the highest among all other reference gestures. Thus, the non-gesture model is proposed which gives a good confirmation for refusing the non-gesture patterns. According to the property of HMM's internal segmentation, the self-transition for each state represents a line-segmented pattern of a gesture path and the outgoing transition from states lead to the rest of sequential segmented patterns in a gesture. Using this property, a model so-called Ergodic is created in which its states are copied from all gesture references in the system and then fully connect these states (Fig. 6.4). The number of states increases as the number of gesture references increases. As a result, the number of edges grows and soon the system becomes unreliable. Therefore, a well-known method (i.e. this method considers all future possibilities for the expansion of the proposed system if the number of reference gestures is increased) is to use the topology of Fig. 6.4(b) where two dummy states are included to make the structure simple. The dummy states (i.e. null states) are nothing and observe no symbol with no time delay. The non-gesture model is constructed by copying the states of all gesture models in the system as follows;

1. Copy all states from all gesture models, each with an output observation probability $b_j(m)$. Then, re-estimate the probabilities with gaussian distribution smoothing filter to make the states represent any pattern. After that, the floor smoothing is applied.

$$Non-gesture(b_j(m)) = \frac{1}{\sqrt{2\pi}\sigma} \cdot exp\left(\frac{(b_j(m))^2}{2\sigma^2}\right) \quad (6.1)$$

2. The probabilities of self-transitions are copied as in the gesture models because each state represents a primitive unit (i.e. segmented pattern) of a gesture. The number of these units constitute the target gesture.

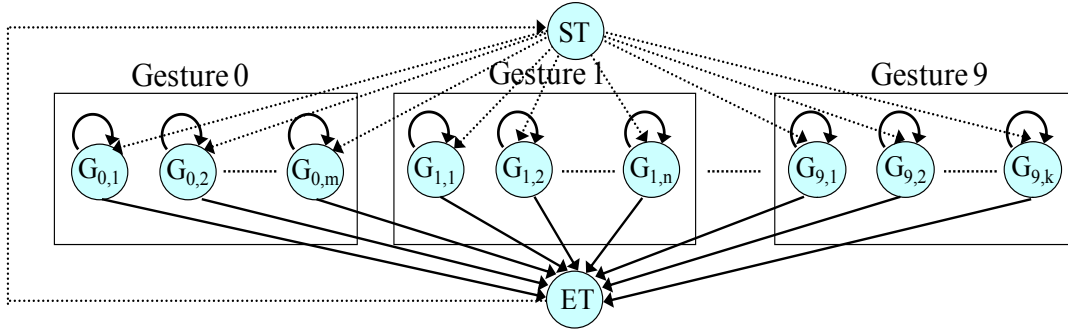


Figure 6.5: The general non-gesture model where the dotted arrows represent null transitions, $G_{i,j}$ refers to the state j for gesture number i , ST and ET are the two dummy states for starting and ending, respectively.

3. The probabilities of all outgoing transitions are calculated as follow;

$$\hat{a}_{ij} = \frac{1 - a_{ij}}{N - 1}, \quad \text{for all } j, i \neq j \quad (6.2)$$

where \hat{a}_{ij} represents the transition probabilities of non-gesture model from state s_i to state s_j , a_{ij} is the transition probabilities of gesture models from state s_i to state s_j and N in the number of states in all gesture models.

The non-gesture model (Fig. 6.5) is a weak model for all trained gesture models and represents every possible pattern. In addition, the likelihood of non-gesture model is smaller than the dedicated model for a given gesture because of the reduced forward transition probabilities. Also, the likelihood of the non-gesture model gives a confidence measure for the calculated likelihood by other gesture models because a confidence measure is based on the differential probability value. This value represents the difference between the observation probability of maximal gestures models and non-gesture model for an input pattern. Thereby, the confidence measure is used as an adaptive threshold for choosing the desired gesture model or gesture spotting.

6.1.3 Model Reduction

The number of states in the non-gesture model is equal to the sum of all states for all gesture models except the two dummy states. This means the number of states for non-gesture model increases as the number of gesture model increases. Furthermore, an increase in the number of states does not affect the recognition rate, but dues to a waste of time and space. To alleviate this problem, relative entropy [105] is used to reduce the non-gesture model states because there are many states with similar probability distribution. The relative entropy is a measure of the distance between two probability distributions. Consider two random probability distributions

$P = p_1, p_2, \dots, p_M$ and $Q = q_1, q_2, \dots, q_M$, the symmetric relative entropy $D(P\|Q)$ is defined as;

$$D(P\|Q) = \frac{1}{2} \sum_{i=1}^M (p_i \cdot \log \frac{p_i}{q_i} + q_i \cdot \log \frac{q_i}{p_i}) \quad (6.3)$$

The proposed state reduction is based on Eq. 6.3 and works as follows;

1. Calculate the symmetric relative entropy between each probability distribution pair $p^{(l)}$ and $q^{(k)}$ of l and k states, respectively.

$$D(P^{(l)}\|Q^{(k)}) = \frac{1}{2} \sum_{i=1}^M (p_i^{(l)} \cdot \log \frac{p_i^{(l)}}{q_i^{(k)}} + q_i^{(k)} \cdot \log \frac{q_i^{(k)}}{p_i^{(l)}}) \quad (6.4)$$

2. Determine the state pair (l, k) with the minimum symmetric relative entropy $D(P^{(l)}\|Q^{(k)})$.
3. Recalculate the probability distribution output by merging these two states over the M observation discrete symbol as;

$$p_i^{(l)*} = \frac{p_i^{(l)} + q_i^{(k)}}{2} \quad (6.5)$$

4. If the number of states is greater than a threshold value with 22 states empirically for spotting system, then go to 1, else re-estimate probability distribution output by gaussian smoothing filter to make the states represent any pattern.

The discrimination of input pattern is computationally expensive when the number of states for non-gesture model is increased. The main advantage of using relative entropy is to reduce the number of states which constitutes the non-gesture model. Thus, the speed of computational process is increased as well as reducing the time and space.

6.1.4 Gesture Spotting Network

In continuous hand motion, key gestures appear intermittently with pre- and post-gestures (i.e. transition for connecting key gestures). To spot these key gestures, gesture spotting network is constructed as shown in Fig. 6.6. Moreover, the gesture spotting network can be easily expanded the vocabularies by adding a new key gesture HMM model and then rebuilding a non-gesture model. This network contains ten gesture models for numbers from 0 to 9. These ten model are designed using LRB model with number of states ranging from 3 to 5 based on its complexity. Additionally, it also contains non-gesture model after states reduction by relative entropy measure and the dummy start state S. The gesture spotting network finds the start and the end points of key gestures which are embedded in the input video stream and performs the segmentation and the recognition tasks simultaneously.

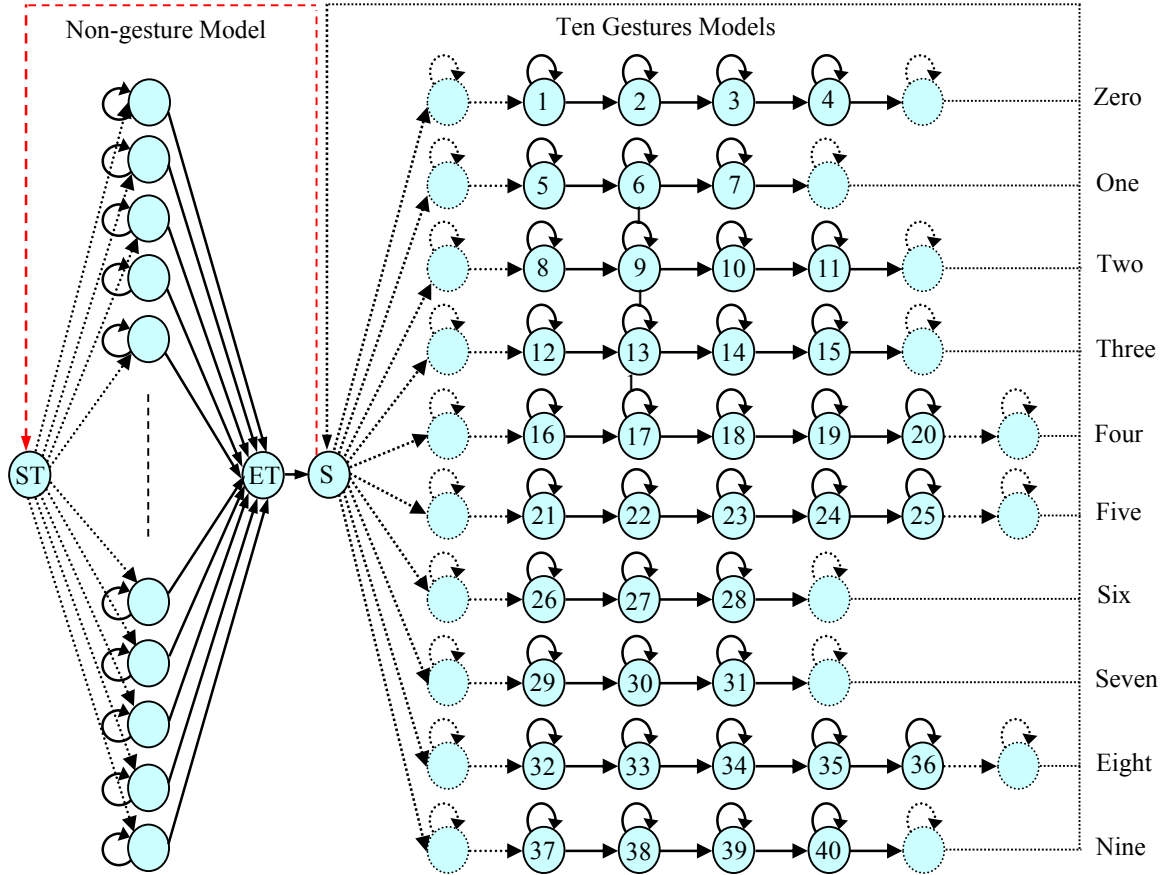


Figure 6.6: The gesture spotting network which contains ten number gesture models from 0 to 9 and are designed by using LRB model with varying states from 3 to 5 and the Non-gesture model.

6.1.5 Spotting and Recognition

For forward spotting, a differential probability (DP) value is defined by the difference between the observation probability value of maximal gesture models and non-gesture model (Fig. 6.7). The maximal gesture is defined as a gesture having the largest value among all ten gestures $p(O|\lambda_g)$ (g is the index of gesture models from 0 to 9). The transition from non-gesture to gesture occurs when the DP value changes from negative to positive (Eq. 6.6, where O is possibly as gesture g). Similarly, the transition from gesture to non-gesture occurs at the time when the DP value changes from positive to negative (Eq. 6.7, where O cannot be a gesture). Consequently, these observation are employed as a rule to detect the start and the end point of gestures. Here, the DP value represents an adaptive threshold which is used for selecting the desired gesture model or gesture spotting.

$$\exists g : P(O|\lambda_g) > P(O|\lambda_{non-gesture}) \quad (6.6)$$

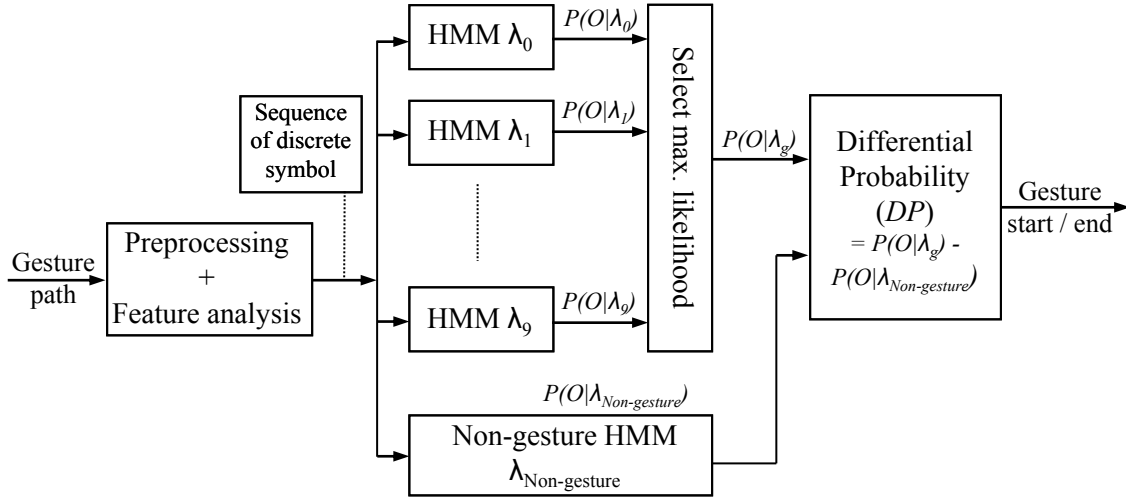


Figure 6.7: Simplified structure showing the main modules for hand gesture spotting via HMMs, where the start and end points are based on differential probability value.

$$\forall g : P(O|\lambda_g) < P(O|\lambda_{non-gesture}) \quad (6.7)$$

The proposed gesture spotting system contains two main modules: segmentation module (segmentation module is also called spotting module) and recognition module. In gesture segmentation module, a sliding window technique is used. This technique calculates the observation probability of all gesture models and non-gesture model for observed segmented parts to spot the start point by DP value. The sliding window (Sw) contains a number of sequential observations instead of a single observation. (Fig. 6.8). It is used to reduce the impact of observation changes for a short interval which are caused by incomplete feature extraction. The optimal value of sliding window is empirically² determined with value 5 where the system is the best in term of results. The gesture recognition module is activated after detecting the start point from continuous image sequences. The main objective is to perform the recognition process accumulatively for the segmented parts until it receives the end signal of key gesture. Therefore, the type of observed gesture segmentation ($\arg \max P(O|\lambda_g)$) is decided at this point using Viterbi algorithm. Then, the processes of these modules are iterated until no more input stream of gesture images exist. Fig. 6.8 illustrates the work of sliding window and the recognition of observed sequences accumulatively. The next steps demonstrate the work of Viterbi algorithm on gesture model λ_g when the number of states is N and the length of observation sequence is T .

1. Initialization:

$$\delta_1^g(i) = \pi_i \cdot b_i^g(o_1); \quad for \ 1 \leq i \leq N \quad (6.8)$$

²Multiple experiments have been conducted with a variety of sliding window size from 1 to 8 on the proposed system to empirically conclude the optimal on the outcome of the system.

2. **Recursion** (accumulative observation probability computation):

$$\delta_t^g(j) = \max_i [\delta_{t-1}^g(i) \cdot a_{ij}^g] \cdot b_j^g(o_t); \quad \text{for } 2 \leq t \leq T, 1 \leq j \leq N \quad (6.9)$$

3. **Termination:**

$$P(O|\lambda_g) = \max_i [\delta_T^g(i)] \quad (6.10)$$

where a_{ij}^g is the transition probability from state i to state j , $b_j^g(o_t)$ refers to the probability of emitting o at time t in state j , and $\delta_t^g(j)$ represents the maximum likelihood value in state j at time t .

Input: An observation sequence O with length T

Output: The probability of key gestures which are embedded in the input stream with their start and end points

Initialize the sliding window Sw

Set $t = 0$ % first time of first segmented pattern

$O' = \{\}$ % initialize the key gesture

Compute $DP(t)$

if $DP(t)$ is negative **then**

while $DP(t)$ is negative **do**

 Shift the sliding window one unit % the start point is not detected

$t = t + 1$

 Compute $DP(t)$

end

else

while $DP(t)$ is positive **do**

$O' = O' \cup \{o_t\}$ % the end point is not detected and union all key gesture segments

 Compute recognition task $\arg \max_g p(O'|\lambda_g)$

$t = t + 1$

 Compute $DP(t)$

end

end

end

if more gesture image **then**

 Set $t = t + 1$

 Repeat the algorithm with re-initializing the sliding window at the value t

else

 | Terminate the algorithm

end

end

Algorithm 3: Gesture Spotting and Recognition

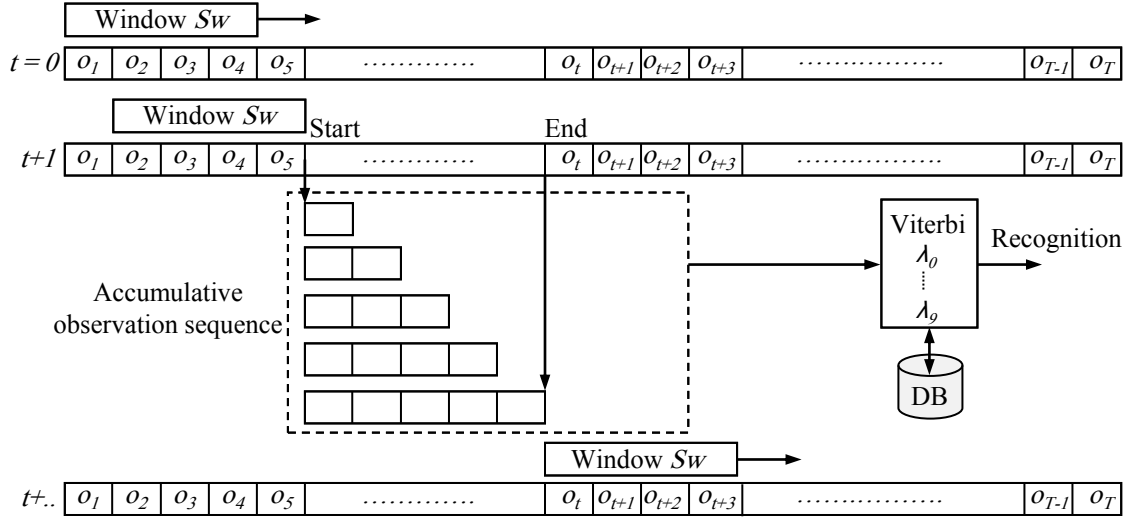


Figure 6.8: Block diagram shows the work of sliding window. The Viterbi algorithm recognizes the segmented parts after detecting the start point.

Assume that, the size of sliding window is Sw and the input observation sequence with length T is $O = \{o_1, o_2, \dots, o_t, \dots, o_T\}$. Firstly, the window size is initialized with an observation sequence $O_{t=0} = \{o_1, o_2, \dots, o_{Sw}\}$. The DP value is equal to difference observation probability between the maximal gesture and the Non-gesture as follows;

$$DP(t) = \max_g P(O_t | \lambda_g) - P(O_t | \lambda_{Non-gesture}) \quad (6.11)$$

When the value of $DP(t)$ is negative, the start point in this case is not detected and therefore the sliding window is shifted on unit (i.e. $O_{t+1} = \{o_{t+1}, o_{t+2}, \dots, o_{Sw+t}\}$). This process is repeated until DP value is positive.

In the case of DP value is positive, assume that O'_1 represents the first partial key gesture segmented. Then, the observed key gesture segmented is represented by union of all possible partial gesture segments $O' = \{O'_1 \cup O'_2 \cup \dots\}$. At each step, the gesture type of O' is determined. When the value of DP becomes negative again or there is no gesture images, the final gesture type g of observed gesture segment O' is determined by Viterbi algorithm. When there are more gesture images, the previous steps are repeated with re-initializing the sliding window at the next time t . Algorithm 3 illustrates the tasks of forward gesture spotting and recognition at the same time according to the sliding window technique.

Thus, the use of HMMs in conjunction with a relative entropy measure and sliding window scheme are capable of modeling spatio-temporal time series of gestures as well as handling non-gesture patterns. In addition, a sophisticated method is proposed for designing a non-gesture model without any training data for non-gesture patterns. Furthermore, forward scheme has the ability to resolve the issues of time delay between gesture spotting and recognition.

6.2 Spotting with CRFs

The key features of CRFs are represented in their conditional nature and the dependency assumptions of their computations to ensure tractable inference. CRFs have all the characteristics of the directed graphical models. In addition, each label in CRFs is employed as exponential model as in MEMMs [26] to conditional probabilities of the next label for a given current label. To spot meaningful gestures of numbers accurately, a stochastic method for designing a non-gesture model with CRFs is proposed. The following subsections explain the stages of spotting and modelling gesture patterns and non-gesture patterns effectively with no training data for non-gesture patterns (Fig. 6.9).

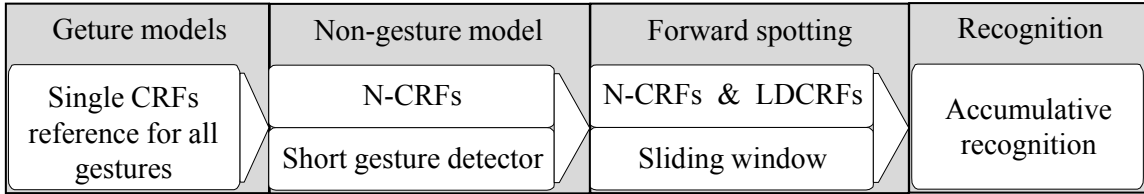


Figure 6.9: Road map of gesture spotting and recognition using CRFs.

6.2.1 Gestures and Non-gesture Model

Conditional Random Fields are undirected graphical models and were developed for labeling sequential data (i.e. determining the probability of a given label sequence for a given input sequence) [16, 26, 136]. The current label is structured to form a chain with an edge between itself and previous label. Moreover, each label corresponds to a gesture number. The probability of label sequence y for a given observation sequence O is calculated as;

$$p(y|O, \theta) = \frac{1}{Z(O, \theta)} \cdot \exp \left(\sum_{i=1}^n F_{\theta}(y_{i-1}, y_i, O, i) \right) \quad (6.12)$$

where in the parameter $\theta = (\lambda_1, \lambda_2, \dots, \lambda_{N_f}; \mu_1, \mu_2, \dots, \mu_{N_g})$, N_f represents the number of transition feature function, N_g refers to the number of state feature function and n is the length of observation sequence O . F_{θ} is defined as;

$$F_{\theta}(y_{i-1}, y_i, O, i) = \sum_f \lambda_f t_f(y_{i-1}, y_i, O, i) + \sum_g \mu_g s_g(y_i, O, i) \quad (6.13)$$

where $t_f(y_{i-1}, y_i, O, i)$ is a transition feature function at position i and $i - 1$ (i.e. represents the weight on the transition from label i to label $i - 1$ when the current observation is O). $s_g(y_i, O, i)$ refers to a state feature function at position i (i.e. represents the weight on the label i when the current observation is O). λ_f and

μ_g represent the weights of the transition and state feature functions, respectively. $Z(O, \theta)$ is the normalized factor and is calculated as follows;

$$Z(O, \theta) = \sum_y \exp \left(\sum_{i=1}^n F_{\theta}(y_{i-1}, y_i, O, i) \right) \quad (6.14)$$

Because CRFs use a single model for the joint probability of the sequences $p(y|O, \theta)$, they are initially built without label for non-gesture pattern. Moreover, CRFs are constructed using combined features of location, orientation and velocity as described in Section 4.3. In addition, CRFs are trained using gradient ascent with the BFGS³ optimization technique [26] with 300 iteration to achieve optimal convergence. Therefore, the labels of CRFs are $y = \{Y_0, Y_1, \dots, Y_9\}$.

All other patterns than gesture patterns are modeled by adding a label (N) for non-gesture patterns to create the Non-gesture model (N-CRFs) using the weights of transition and state features function of initial CRFs. Moreover, the labels of N-CRFs are $y_N = \{Y_0, Y_1, \dots, Y_9, Y_N\}$. The proposed N-CRFs model does not need non-gesture patterns for training and also can better spot gestures and non-gesture patterns.

6.2.2 N-CRFs Model Parameters

There are two main parameters of CRFs named *state feature function* and *transition feature function* as in Eq. 6.13. By using the weight of state and transition feature function of the initialized CRFs model, the label of non-gesture pattern is created. From the idea of Dugad *et al.* [137] to propose an adaptive threshold model based on the mean and the variance of sample, the weight of state feature function is computed as;

$$\mu_g(N) = \bar{\mu}_g + T_N \sqrt{\sigma_g} \quad (6.15)$$

where $\bar{\mu}_g$ is the mean of state feature functions of the labels of initial CRFs from Y_0 to Y_9 and σ_g represent the variance of the g^{th} state feature functions. T_N reflects the width of state features function in some way. The optimal value of T_N is 0.7 and is determined by multiple experiments which have been conducted with a range of values on a training data set.

A challenging problem is caused by the fact that there is a quite bit of variability in the same gesture even for the same person. The main advantage of HMMs is its capability of modeling spatio-temporal time series of gestures effectively. Whereas this problem is one of difficulties which are faced for CRFs in recognition. In other words, it is difficult to spot and recognize short gestures because short gestures have

³Broyden-Fletcher-Goldfarb-Shanno (BFGS) technique is used to solve nonlinear optimization problems which have lack constraints to determine the best parameter values by using the smallest number of computational iterations. This technique uses the first and second derivatives so that the gradient with zero represents a necessary condition for optimality.

fewer samples than long gestures. To avoid this problem, a short gesture detector is added where the weights of self-transition feature functions are increased as follows;

$$\lambda_f(Y_l, Y_l) = \begin{cases} \lambda_f(Y_l, Y_l) + \psi_f(Y_l), & \text{if } N_{frame}(Y_l) < (\bar{N}_{frame} - \sigma_{N_{frame}}) \\ \lambda_f(Y_l, Y_l), & \text{otherwise} \end{cases} \quad (6.16)$$

and

$$\psi_f(Y_l) = \frac{(\bar{N}_{frame} - \sigma_{N_{frame}}) - N_{frame}(Y_l)}{\max_l N_{frame}(Y_l)} \quad (6.17)$$

where $N_{frame}(Y_l)$ is the average frame number of a gesture Y_l , \bar{N}_{frame} represents the average frame number of all gestures from Y_0 to Y_9 and $\sigma_{N_{frame}}$ is the variance of them. $\psi_f(Y_l)$ is additional weight of the gesture Y_l notable in case of a short length gesture. Fig. B.1 in Appendix B illustrates the average number of frames for each isolated gesture from 0 to 9.

The weight of the self-transition feature function of the label of non-gesture patterns is approximately assigned with the maximum weight of transition feature functions to initialize CRFs as follows;

$$\lambda_f(Y_N, Y_N) = \max_l \lambda_f(Y_l, Y_l) + \frac{\sum_{i=1}^l \sum_{g=1}^{N_g} \mu_g(Y_l)}{\bar{N}_{state-feature}} \quad (6.18)$$

where $\bar{N}_{state-feature}$ is the average number of transition feature functions in which the weight is greater than zero.

As described earlier in this chapter about the transition parameters of non-gesture model via HMMs, nearly a similar method is employed to compute the weights of transition feature functions between the labels of gesture models and the label of non-gesture patterns. Therefore, the weights of transition feature functions from the non-gesture label to other labels are computed by the following equation;

$$\lambda_f(Y_N, Y_i) = \frac{\lambda_f(Y_N, Y_N)}{l}, \quad \forall i \in \{1, 2, \dots, l\} \quad (6.19)$$

Also, the weights of transition feature functions from the gesture labels to non-gesture label occurs by the given equation below;

$$\lambda_f(Y_i, Y_N) = \frac{\lambda_f(Y_i, Y_i)}{l}, \quad \forall i \in \{1, 2, \dots, l\} \quad (6.20)$$

Thus, the N-CRFs model can better spot gestures and non-gesture patterns.

6.2.3 Forward Gesture Spotting and Recognition

In order to spot and recognize the key gestures from continuous image sequences, the two main modules are applied: segmentation module and recognition module.

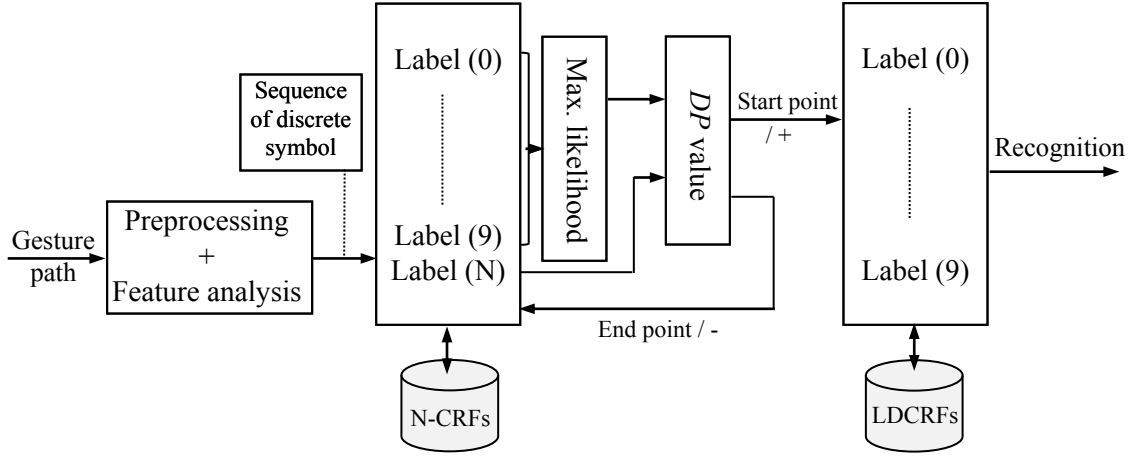


Figure 6.10: Simplified structure showing the main modules for hand gesture spotting via CRFs.

In gesture segmentation module, the sliding window Sw with empirically estimated optimal size of 5 is employed to calculate the observation probability of all gesture labels and non-gesture label for the segmented parts using N-CRFs database. The sliding window technique was described in Section 6.1.5 (Fig. 6.8). The start (end) point of gesture is spotted by differential probability DP value between maximal gestures labels and non-gesture label. When DP value changes from negative to positive, the gesture starts. Similarly, the gesture ends at the time when the DP value changes from positive to negative. These observations are employed as a rule to detect the start and the end points of meaningful gestures. When the DP value is negative, the process is repeated until the start point of key gesture is detected. Spotted gesture in the segmentation module are temporarily saved. Therefore, the correct spotting is defined as;

$$Correct_{spotting} = \begin{cases} true, & \text{if } |Start_{frame} - End_{frame}| \geq \epsilon \\ false, & \text{otherwise} \end{cases} \quad (6.21)$$

where ϵ is the length of a short gesture path. Moreover, in proposed gesture spotting system, a gesture path '1' has a short length in which the minimum number of frames assigned to it is equal to $\epsilon = 12$ frames (Fig. B.1).

The gesture recognition module using LDCRFs database⁴ is activated after detecting the start point from continuous image sequences to recognize the segmented parts until it receives the end signal of key gestures. Then, the processes of these

⁴LDCRFs database is prepared according to 420 video samples for isolated gestures and are captured from three subjects on a set of numbers. LDCRFs model is trained using gradient ascent with the BFGS optimization technique.

modules are iterated until no more input stream of gesture images exist (Fig. 6.10). Here, LDCRFs model is considered because the recognition using LDCRFs is the best in terms of result than CRFs. For more details about the comparison between CRFs and LDCRFs according to their recognition results, the reader can refer to Section 5.3.3. In addition, Algorithm 3 is applied using the CRFs model with the same tasks similar to HMMs.

6.3 Computational Complexity

In HMMs case, the adequate number of states for Non-gesture model is nearly equal to one and a half number of observation symbols through a set of experiments. Thereby, the time complexity C is computed which is employed to evaluate gesture spotting as follows;

$$C = La\bar{N}T + N_{ng}^2T \quad (6.22)$$

where L is the number of gesture models ($L = 10$), a represents the number of transition per state (the number of transition per state is 2 because of using LRB topology), \bar{N} represents the average number of states of gesture models (i.e. $\bar{N} = 4$) and T refers to the length of observation sequence. In gesture spotting system, the codeword size (i.e. the average number of observation symbols) is nearly equal to fifty and the number of states in the Non-gesture model is decreased from 40 to 22 using relative entropy (i.e. the number of states before reduction $N_{ng} = 40$ and the number of states after reduction is equal to 22). The relative entropy provides a way of reducing the number of states where the increased number of states causes the waste of time and space. Consequently, the expected rate for the reduction of the evaluation time for gesture spotting is;

$$Evaluation\ time = \frac{(La\bar{N}T + N_{ng}^2T) - (La\bar{N}T + \hat{N}_{ng}^2T)}{La\bar{N}T + N_{ng}^2T} \quad (6.23)$$

where \hat{N}_{ng} represents the minimized number of states for the Non-gesture model. The Eq. 6.23 is simplified as follows;

$$Evaluation\ time = \frac{N_{ng}^2 - \hat{N}_{ng}^2}{La\bar{N} + N_{ng}^2} = \frac{40^2 - 22^2}{(10) \cdot (2) \cdot (4) + 40^2} = 0.6642 \quad (6.24)$$

Thus, the expected time saved for the evaluation is 66.42% (Eq. 6.24).

The time complexity of the Viterbi algorithms presented in this chapter is proportional to the number of cells which are visited by dynamic programming method. Each gesture path for numbers takes $O(TN)$ where T is length of the gesture path and N is the number of states for a specific gesture path. The space complexity of the Viterbi matching algorithm is similar to the time complexity if the algorithm is run in offline and online mode.

6.4 Experimental Results and Analysis

The segmentation of the hand with complex background took place using depth map and color information over YC_bC_r color space. Gaussian Mixture Models were considered for this purpose where a large database of skin and non-skin pixel is used for training. Furthermore, morphological operations and Mean-shift algorithm are used to track the hand to generate gesture path. The features of hand gesture path were extracted according to two different locations, three varying orientations and velocity. The extracted features were quantized using k -means clustering algorithm to obtain discrete symbols and applied to HMMs and CRFs models. The input images were captured by Bumblebee stereo camera system which has 6 mm focal length at 15FPS with 240×320 pixels image resolution, Matlab and C++ language implementation.

Classification results are based on our database and it contains 600 video samples for isolated gestures which are captured from three persons on a set of numbers. Each number from 0 to 9 was based on 42 videos for training and 18 video samples for testing (In total, 420 video samples for training and 180 video samples for testing). Also, the database contains 280 video samples of continuous hand motion for testing. Each video sample either contains one or more meaningful gestures. The HMMs have been trained by BW algorithm while the CRFs model was trained using gradient ascent with the BFGS optimization technique with 300 iteration to converge. The inference (i.e. recognition) process uses forward score of each sample to select the model with the highest likelihood. The experiments are carried out for an isolated gesture recognition and key gestures spotting test. The following sections discuss the analysis of HMMs and CRFs results in details.

6.4.1 Key Gesture Spotting with HMMs

The gesture recognition module match the tested gesture against database of reference gestures to classify which class it belongs to. The higher priority was computed by Viterbi algorithm to recognize the numbers frame by frame using LRB topology with different number of states ranging from 3 to 5 based on their complexity. Fig. 6.11 illustrates the result of isolated gesture ‘3’ with high four priorities while the probability of Non-gesture model before and after state reduction is the same. Moreover, the number of states of Non-gesture model before state reduction is 40 and after reduction is 22 state. This in turn leads to several advantages such as saving time and space and most importantly, makes the system appropriate to real-time applications. From table 6.1, the recognition ratio of isolated gestures achieves best results with 97.78%. The recognition ratio ($Rec.$) is the number of correctly (true) recognized gestures to the number of tested gestures (Eq. 6.25).

$$Recognition\ ratio = \frac{\# recognized\ gestures}{\# test\ gestures} \times 100 \quad (6.25)$$

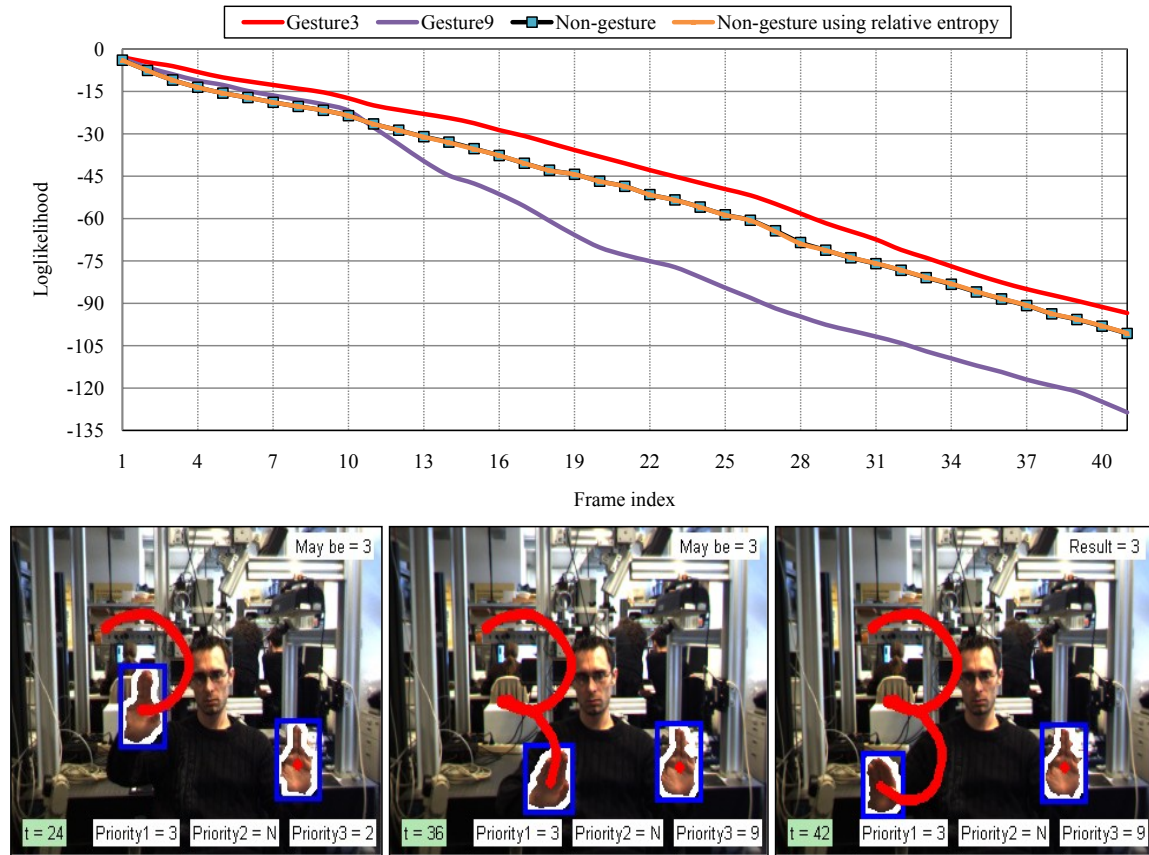


Figure 6.11: Temporal evolution of four higher probabilities of the gestures ‘3’, ‘9’, ‘Non-gesture’ before and after state reduction. The probability of Non-gesture model before and after state reduction is the same. In the image sequences, the high priority is gesture ‘3’ and the second priority refers to Non-gesture ‘N’ at $t = 24$. The final result is gesture number ‘3’ at $t = 42$.

In automatic gesture spotting task, there are three types of errors called insertion (I), substitution (S) and deletion (D).

The insertion error is occurred when the spotter detects a nonexistent gesture. It is because the emission probability of the current state for a given observation sequence is equal to zero. A substitution error occurs when the key gesture is classified falsely (i.e. classifies the gesture as another gesture). This error is usually happened when the extracted features are falsely quantized to other codewords. The deletion error happens when the spotter fails to detect a key gesture. In order to calculate the recognition ratio (Eq. 6.25), insertion errors are totally not considered. However, insertion errors are probably caused due to substitution and deletion errors because they are often considered as strong decision in determining the end point of gestures to eliminate all or part of the meaningful gestures from observation. Deletion errors directly affect the recognition ratio whereas insertion errors do not. However, the

Table 6.1: Isolated gesture recognition and key spotting gesture results for gesture numbers from '0' to '9' using HMMs at sliding window equal to 5.

Gesture path	Train data	Isolated gestures results			Key gestures spotting results					
		Test	correct	Rec.(%)	Test	I	D	S	correct	Rel.(%)
'0'	42	18	17	94.44	28	2	1	2	25	83.33
'1'	42	18	18	100.00	28	0	1	1	26	92.86
'2'	42	18	17	94.44	28	0	0	2	26	92.86
'3'	42	18	18	100.00	28	0	0	0	28	100.00
'4'	42	18	18	100.00	28	0	0	1	27	96.43
'5'	42	18	18	100.00	28	0	1	1	26	92.86
'6'	42	18	17	94.44	28	1	1	1	26	89.66
'7'	42	18	18	100.00	28	0	0	0	28	100.00
'8'	42	18	17	94.44	28	1	0	2	26	89.66
'9'	42	18	18	100.00	28	0	1	0	27	96.43
Total	420	180	176	97.78	280	4	5	10	265	93.31

insertion errors affect the gesture spotting ratio directly. To take into consideration the effect of insertion errors, another performance measure called reliability (*Rel.*) is proposed by the following equation;

$$Rel. = \frac{\# \text{ correctly recognized gestures}}{\# \text{ test gestures} + \# \text{ Inseration errors}} \times 100 \quad (6.26)$$

The recognition ratio and the reliability are computed based on the number of spotting errors (Table 6.2). The gesture spotting accuracy is measured according to different sliding window size ranging from 1 to 8 (Fig. 6.12(a)). Furthermore, the gesture spotting accuracy is improved initially as the sliding window size increase, but degrades as sliding window size increase further. Therefore, the optimal size of sliding window is 5 empirically where the reliability of automatic gesture spotting system achieves 93.31%. In Fig. 6.12(b), the number of errors decreases sharply between $Sw = 1$ and $Sw = 4$. However, deletion, insertion and substitute errors begin to increase after $Sw = 4$. The increase in the size of Sw means that it contains some of observation features belong to gesture patterns and others belong to non-gesture patterns, and hence this leads to loss of starting and ending points of meaningful gestures. Table 6.2 illustrates the recognition rate of isolated and key gestures relative to different window size ranging from 1 to 8. Furthermore, the yield of isolated training data is higher than isolated testing data. In addition, the overall recognition

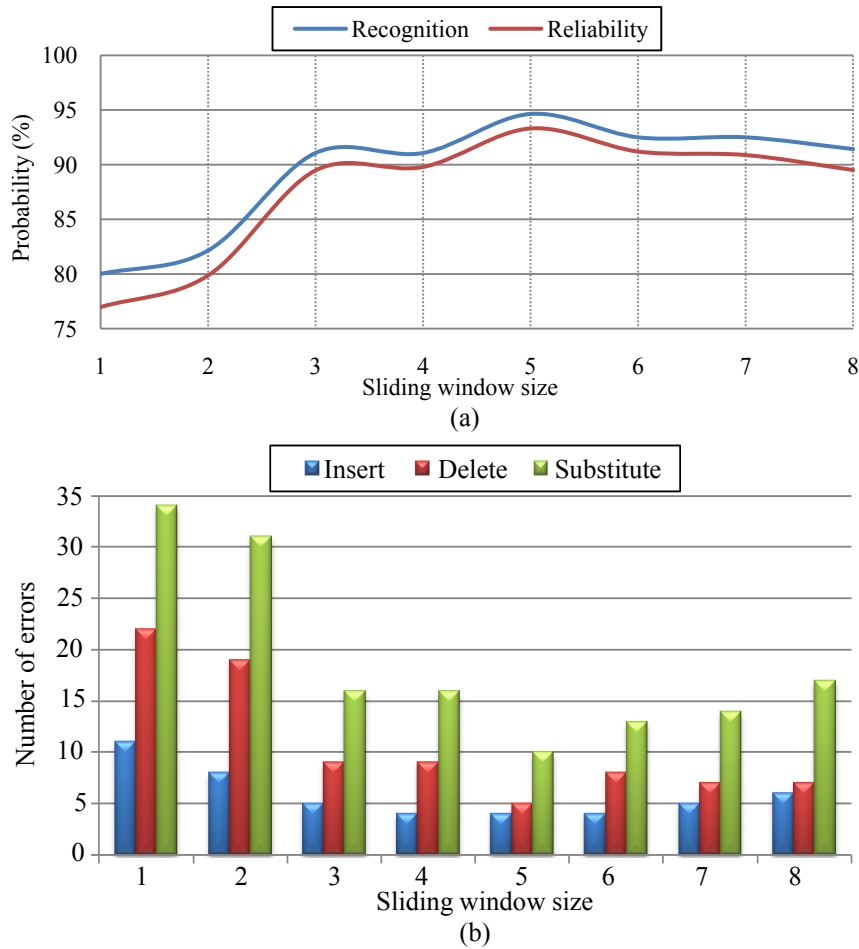


Figure 6.12: (a) Spotting accuracy using HMMs relative to sliding window size ranging from 1 to 8. (b) Insertion, deletion and substitution errors relative to sliding window size.

rate is the average of training and testing recognition rate and achieved 98.35% at the sliding window size equal to 5. Fig. 6.13 shows the results of continuous gesture path which contains one meaningful gestures ‘6’ where the start point at frame index = 15 and the end point at frame index = 50. Moreover, the proposed system automatically recognizes isolated and key hand gestures with superior performance and low computational complexity when applied to several video samples containing confusing situations such as occlusion between hands and face. Experimental results of HMMs show that the proposed system automatically recognizes isolated gestures with 97.78% and key gestures with 93.31% reliability. It is noted that the proposed system achieved high recognition rate for isolated gestures and is due to a good election for the set of feature candidates to optimally discriminate among input patterns. Also, a careful experimental based selection of initialization parameters

Table 6.2: Results of isolated gestures recognition and key gestures spotting with different size of sliding window (Sw) ranging from 1 to 8 via HMMs.

Sw	Train data	Isolated gestures results				Spotting key gestures results					
		Test data	Recognition (%)			Test data	Error types			Spotting (%)	
			Train	Test	Overall		I	D	S	Rec.	Rel.
1	420	180	86.79	86.11	86.45	280	11	22	34	80.00	76.98
2	420	180	89.29	87.78	88.53	280	8	19	31	82.14	79.86
3	420	180	92.50	91.11	91.81	280	5	9	16	91.07	89.47
4	420	180	95.00	92.22	93.61	280	4	9	16	91.07	89.79
5	420	180	98.93	97.78	98.35	280	4	5	10	94.64	93.31
6	420	180	96.07	93.89	94.98	280	4	8	13	92.50	91.20
7	420	180	95.71	94.44	95.08	280	5	7	14	92.50	90.88
8	420	180	95.35	94.44	94.90	280	6	7	17	91.43	89.51

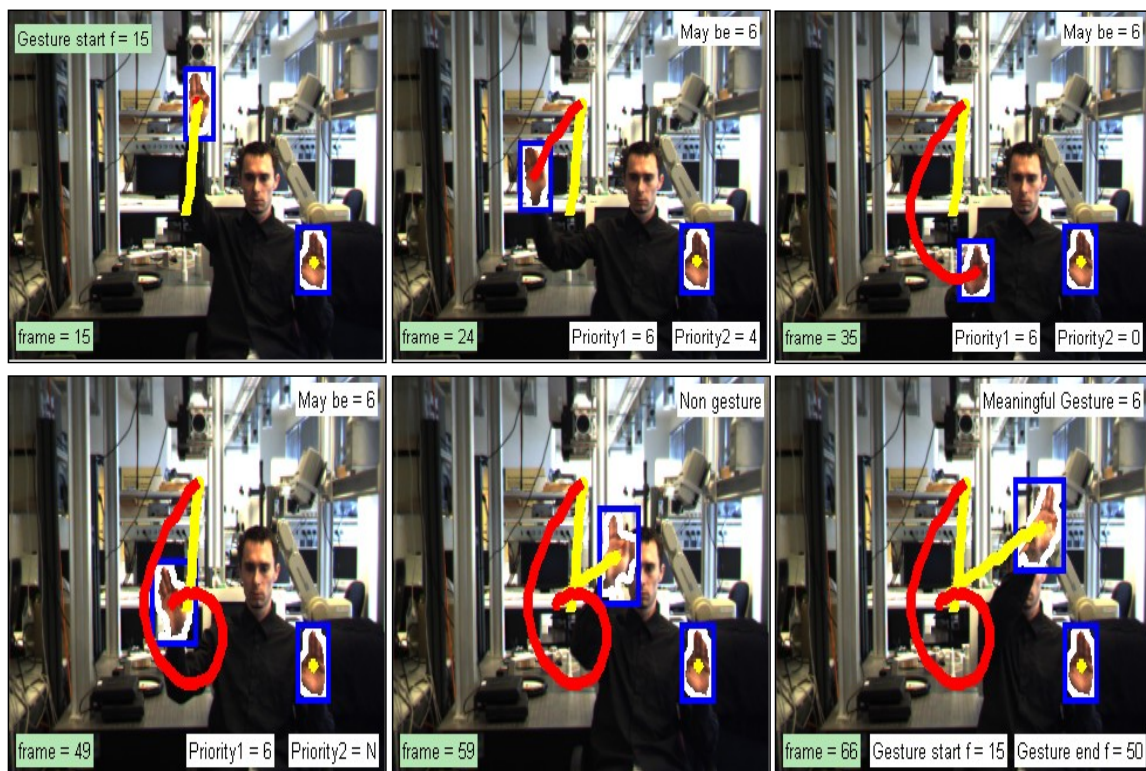


Figure 6.13: Image sequences contain one meaningful gesture ‘6’, where the start point at frame 15 and the end point at frame 50. ‘N’ refers to Non-gesture.

for the training process. In addition, HMMs have the ability to efficiently alleviate spatio-temporal variabilities. Thus, this system is capable for real-time applications and resolves the issues of time delay between spotting and recognition tasks.

6.4.2 Key Gesture Spotting with CRFs

CRFs are constructed using the combined features of location, orientation and velocity as described in Section 4.3. On a standard desktop PC, training process is more expensive for CRFs since the time which the model needs ranges from 20 minutes to several hours based on observation window. On the contrary, the inference (i.e. recognition) process is less costly and very fast for all models with sequences of several frames. Moreover, CRFs run at window size = 2 which means that the input feature vector at each frame consists of five feature vectors; the current frame, the two preceding frames and the two future frames.

Table 6.3 shows the recognition rate of isolated gestures according to the training and the testing data with different sizes of sliding window ranging from 1 to 8. The yield of training data is higher than testing data because the training data represents the reference of the proposed system. The gesture recognition rate is improved initially as the window size increase, but degrades as window size increase further. The increase in window size leads to loss of starting and ending points of meaningful gestures. Therefore, the optimal window size is chosen as 5 empirically, where the proposed method automatically recognizes tested gestures with 94.44%. In addition, the overall recognition rate achieved is 96.51%.

As described in previous section, there are three types of errors called insertion, substitution and deletion. Deletion errors directly affect the recognition ratio whereas insertion errors do not. Here, it is to be noted that, the insertion errors affect the gesture spotting ratio directly since they are probable to cause substitution and deletion errors. These errors are estimated, and then the recognition ratio and the reliability are calculated (Table 6.3). The gesture spotting accuracy is measured according to different sliding window sizes ranging from 1 to 8 (Fig. 6.14(a)). It noted that the gesture spotting accuracy is improved initially as the sliding window size increase, but degrades as sliding window size increase further. Therefore, the optimal sliding window size = 5 where multiple experiments have been conducted with a variety of sliding window size on the proposed system to empirically conclude the optimal on the outcome of the system. The reliability of CRFs spotting system is improved from 86.12% to 90.49% using a short gesture detector (Table 6.3). In Fig. 6.14(b), the number of errors decreases sharply between $Sw = 1$ and $Sw = 4$. However, deletion, insertion and substitute errors begin to increase after $Sw = 4$.

Fig. 6.15 illustrates a gesture spotting result for image sequences which are depicted in Fig. 6.16. The label of non-gesture has the greatest probability during the first 19 frames. Then, it is followed by the gesture '3' until frame number 51. After

Table 6.3: Results of isolated gestures recognition and key spotting gestures with different size of sliding window (Sw) ranging from 1 to 8 using CRFs.

Sw	Isolated gestures results				Test data	Spotting key gestures without short gesture detector				Spotting key gestures with short gesture detector								
	Train		Test			Error types		Spotting (%)		Error types		Spotting (%)						
	data	Recognition (%)	data	Overall		I	D	S	Rec.	Rel.	I	D	S	Rec.	Rel.			
1	420	180	90.48	83.33	86.33	280	12	30	40	210	75.00	71.92	10	24	38	218	77.86	75.17
2	420	180	93.10	86.11	89.60	280	10	25	36	219	78.21	75.52	7	20	33	227	81.07	79.09
3	420	180	93.57	87.22	90.40	280	9	21	28	231	82.50	79.93	6	16	25	239	85.36	83.56
4	420	180	95.48	90.56	93.02	280	9	15	23	242	86.43	83.74	4	9	18	253	90.36	89.08
5	420	180	98.57	94.44	96.51	280	8	11	21	248	88.57	86.12	4	7	16	257	91.79	90.49
6	420	180	97.62	91.11	94.37	280	10	13	20	247	88.21	85.17	5	9	19	252	90.00	88.42
7	420	180	97.62	92.22	94.92	280	10	12	21	247	88.21	85.17	5	11	18	251	89.64	88.07
8	420	180	96.90	89.44	93.17	280	11	13	24	243	86.79	83.51	5	10	21	249	88.93	87.37

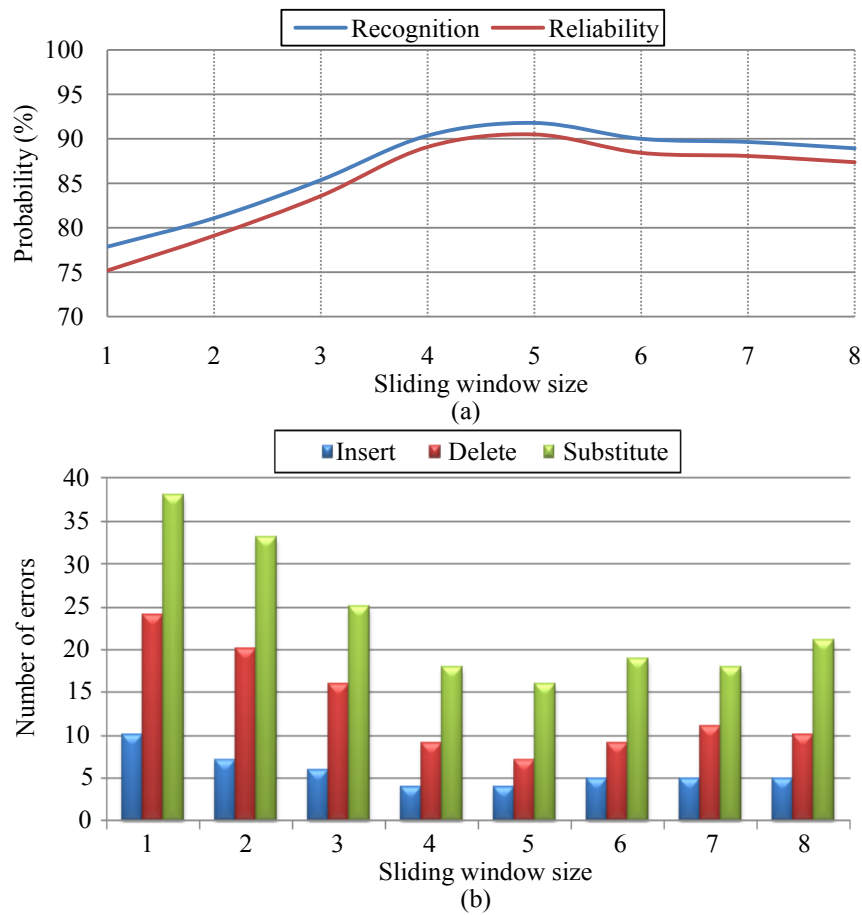


Figure 6.14: (a) Spotting accuracy using CRFs relative to sliding window sizes (1-8). (b) Insertion, deletion and substitution errors relative to sliding window size.

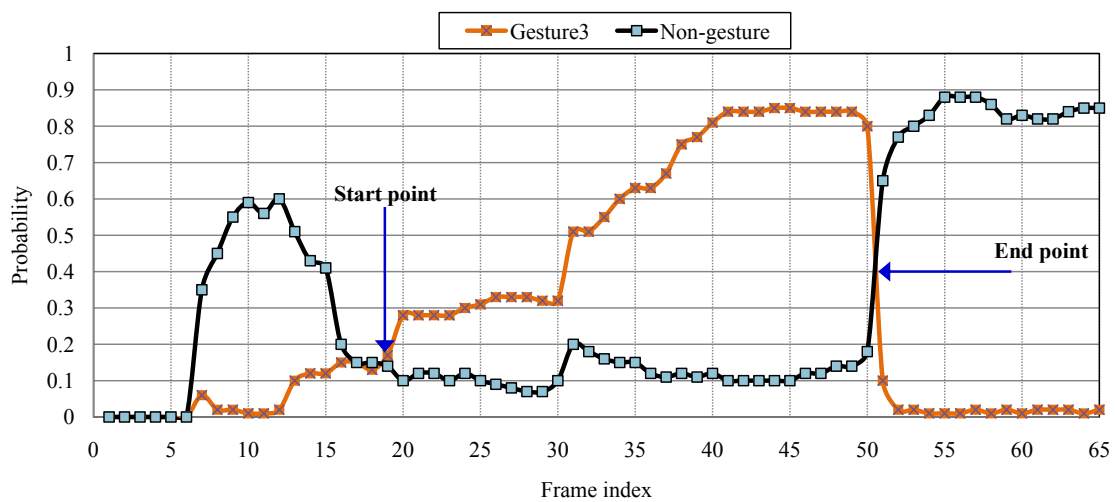


Figure 6.15: Temporal evolution of gesture '3' and non-gesture probabilities.

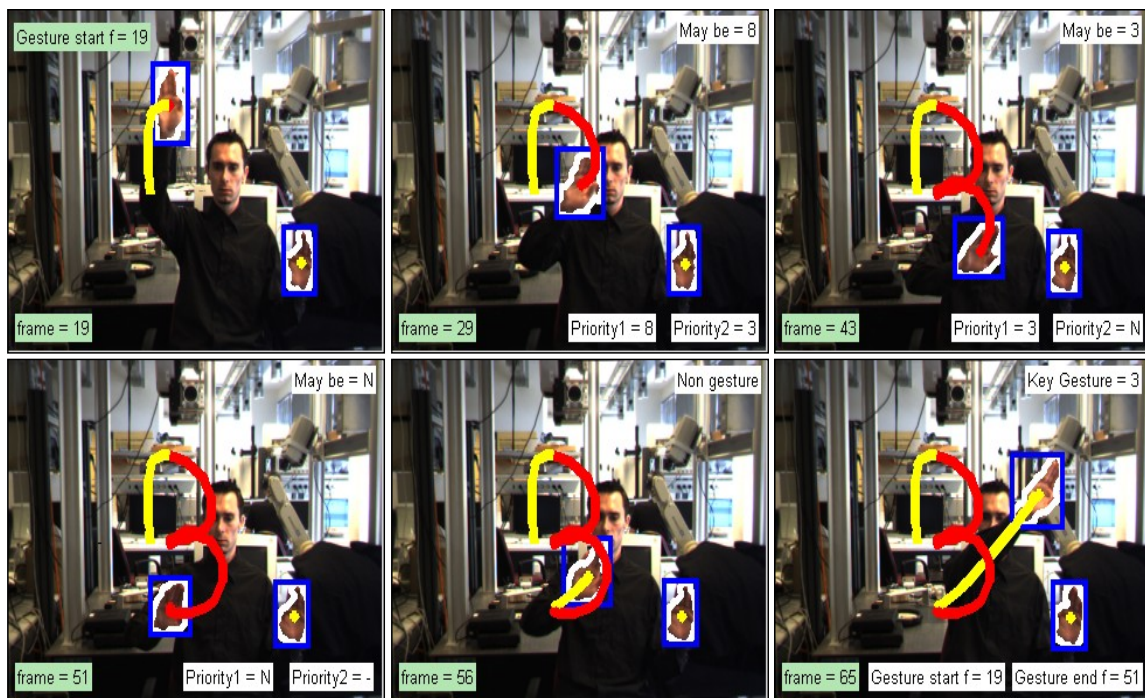
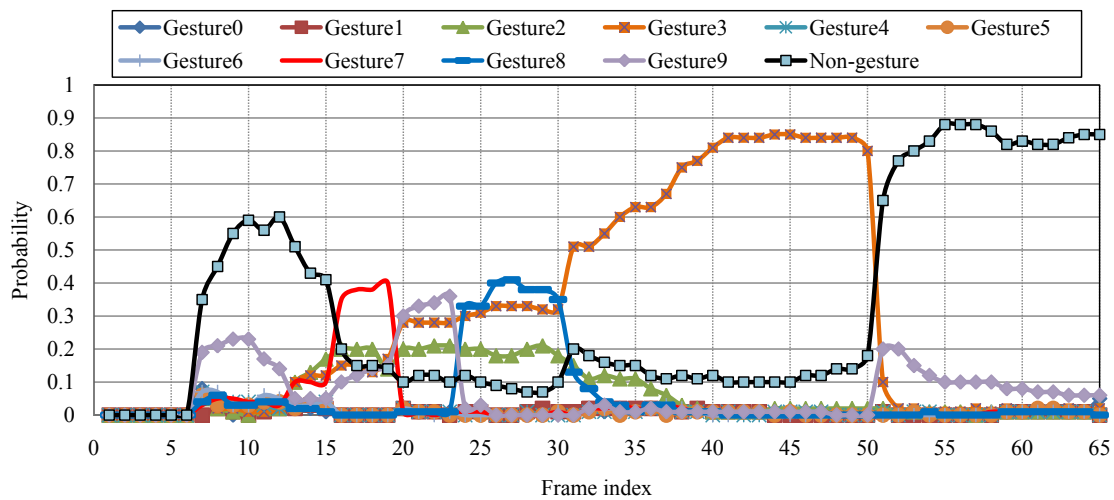


Figure 6.16: Temporal evolution of the probabilities of the gesture numbers (0-9) and non-gesture label 'N'. The image sequences contain one key gestures '3', where the start point is at frame 19 and the end point is at frame 51. In the first 18 frames, the probability of non-gesture label is assigned the greatest value, which means that the start point of the key gesture is not detected. At frame 19, the start point is detected since the higher priority is assigned to gesture labels than the non-gesture label. At frame 51, the higher priority is non-gesture label which means that the end point of key gesture '3' is detected.

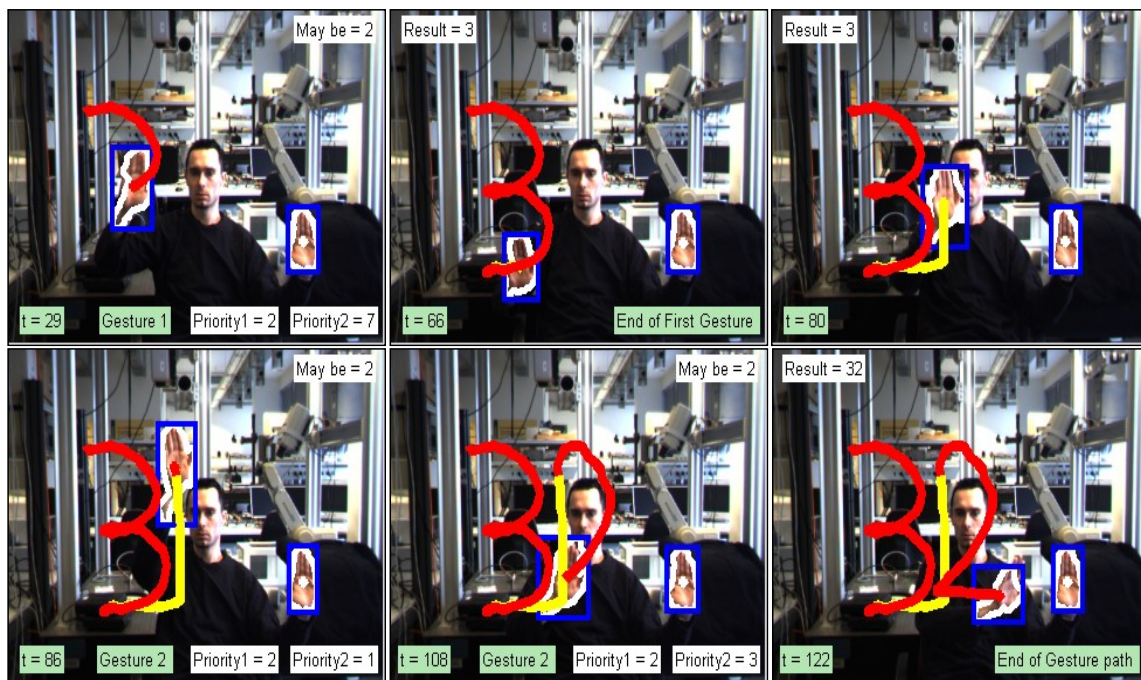
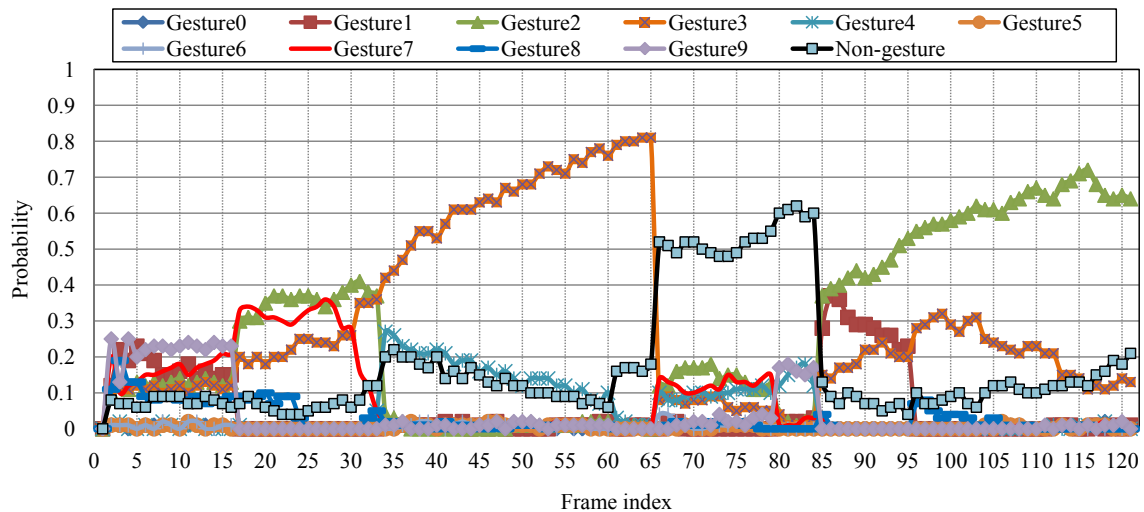


Figure 6.17: Temporal evolution of the probabilities of the gesture numbers (0-9) and non-gesture label 'N'. The image sequences contain two key gestures '3', '2', where the end point of meaningful gesture '3' at frame 66 and the start point of meaningful gesture '2' at frame 85. Between frame 67 and frame 85, the higher priority is assigned to non-gesture label which means that the start point of the second key gesture is not detected. At frame 86, a new key gesture is started where the probability value of non-gesture label is not the highest value as compared to the other gesture labels.

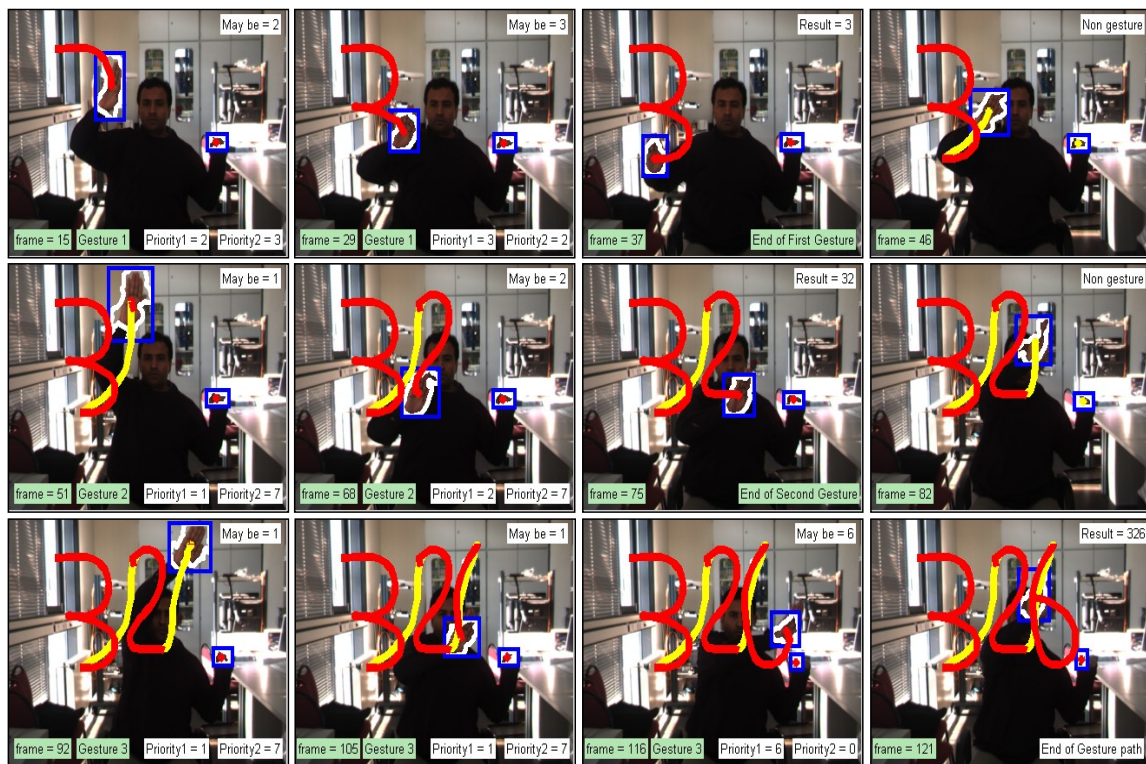
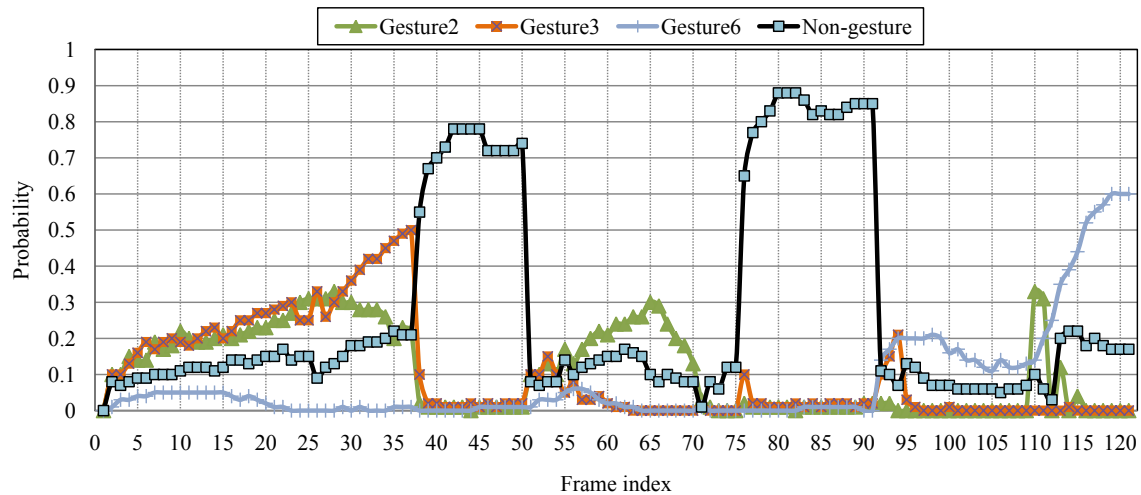


Figure 6.18: Temporal evolution of the probabilities of the gesture numbers ‘2’, ‘3’, ‘6’ and non-gesture label ‘N’. The image sequences contain three key gestures ‘3’, ‘2’, ‘6’. The end point of meaningful gesture ‘3’ is at frame 37. Between frame 37 and frame 50, the higher priority is assigned to non-gesture label which means that the start point of the second key gesture is not detected. At frame 51, a new key gesture has started where the probability value of non-gesture label is not the highest value as compared to the other gesture labels. The end point of meaningful gesture ‘2’ is at frame 75. Between frame 75 and frame 91, the higher priority is assigned to non-gesture label. The start point of meaningful gesture ‘6’ is at frame 92. The final result of the continuous gesture path is ‘326’.

51 frames, the transition of non-gesture increase again. Therefore, the end point is detected at frame index 51. Fig. 6.17 shows the temporal evolution of the probabilities of the gestures number (0-9) and non-gesture. The image sequence contains two key gestures ‘3’, ‘2’, where the gesture ‘3’ ends at frame index 66 and the start point of gesture ‘2’ is at frame index 85. The image sequences depicted in Fig. 6.18 contain three key gestures ‘3’, ‘2’ and ‘6’. The above graph of this figure considers only the temporal evolution of the probabilities of gestures ‘2’, ‘3’, ‘6’ and non-gesture (for simplicity, the other curves are eliminated because their probabilities are low). The gesture ‘3’ ends at frame index 37. Between frame index 37 and frame index 50, the higher priority is assigned to non-gesture label which means that the start point of second key gesture is not detected. At frame index 51, a new key gesture is started where the probability value of non-gesture label is not the highest value as compared to the other gesture labels. The gesture ‘2’ ends at frame index 75. Between frame index 75 and frame index 91, the higher priority is assigned to non-gesture label. The gesture ‘6’ starts at frame index 92 and ends at frame index 121. Moreover, the proposed system automatically recognizes isolated and key hand gestures with superior performance and low computational complexity. Additionally, the system has the ability to deal with several video samples which contain confusing situations such as partial occlusion among hands and face. Experimental results with CRFs show that the proposed system automatically recognizes isolated gestures with 94.44% and key gestures with 90.49% reliability. For more results, the reader can refer to Fig. B.5, Fig. B.6, Fig. B.7 and Fig. B.8 in Appendix B.

6.4.3 Gesture Spotting with HMMs versus CRFs

The difference between HMMs and CRFs is that HMMs are generative models which define a joint probability distribution to solve a conditional problem, thus focusing on modeling the observation to compute the conditional probability. Moreover, one HMM is constructed per label (i.e. each gesture number) where HMMs assume that all the observation are independent. Whereas, CRFs are undirected graphical models and were developed for labeling sequential data. In addition, CRFs overcome the weakness of directed graphical models which suffer from the bias problem as in MEMMs [26]. Moreover, CRFs use a single model for all numbers. The HMMs use a Left-Right Banded model based on Gaussian emission probabilities having a full covariance matrix for each state. The HMMs parameters (i.e. the emission probability and the state transition matrix) are learned from the same training data used by CRFs. The HMMs models were trained by BW algorithm while CRFs were trained using gradient ascent with the BFGS optimization technique with 300 iteration to converge. Training process is more expensive for CRFs than HMMs since the required time to model ranges from 20 minutes to several hours based on observation window. On the contrary, the recognition process is less costly and very fast for all

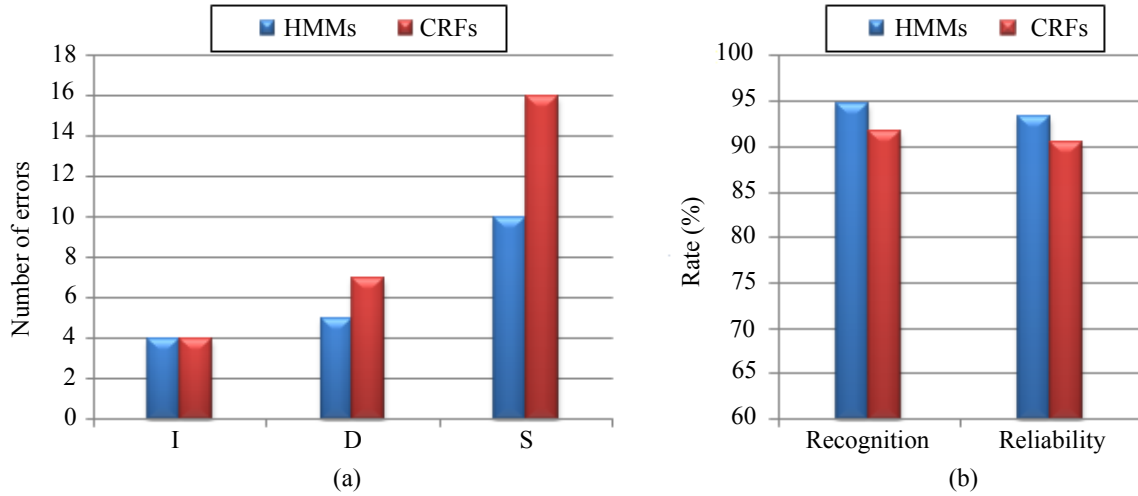


Figure 6.19: A comparison result between HMMs and CRFs. (a) Error types (Insertion: I, Deletion: D, substitution: S) of CRFs and HMMs. (b) The recognition and the reliability of HMMs and CRFs where the reliability of system considers the insertion error in calculation.

models with sequences of several frames. The type of observed gesture is decided with HMMs by Viterbi algorithm frame by frame.

Fig. 6.19 and Table 6.4 show the comparison between HMMs and CRFs at sliding window size = 5. The results show that the proposed system successfully spots and recognizes key gestures with 93.31% and 90.49% reliability for HMMs and CRFs respectively. In addition, the number of deletion and substitution errors for CRFs are higher than its own ideals of HMMs. Whereas the insertion error is the same for both HMMs and CRFs. In general, the proposed HMMs is the best in term of results for spotting gestures than CRFs. After the reduction of the states in Non-gesture model using HMMs, the model inference is faster and the evaluation time of 66.42% is saved.

The backward spotting techniques firstly detect the end points of gestures and then tracking back through their optimal paths to discover the start points of gestures. Upon the detection of the start and the end points, in-between trajectory is sent to recognizer for recognition. So, there is a time delay between the meaningful gesture spotting and recognition and this time delay is unacceptable for online applications. The main contribution of gesture spotting system was to propose a forward gesture spotting to handle hand gesture segmentation and recognition at the same time. In addition, a stochastic methods for designing a non-gesture model from HMMs and CRFs model with no training data for non-gesture patterns are proposed.

Fig. 6.20 presents the average spotting time of the backward and the forward spotting for each gesture (0-9) at $Sw = 5$. The backward spotting technique takes a

Table 6.4: Results of spotting key gestures using HMMs versus CRFs.

Gesture path	Train		Test		Spotting key gestures using HMMs						Spotting key gestures using CRFs								
	data		data		Error types		correct		Results (%)		Error types		correct		Results (%)				
	I	D	S	I	D	S	I	D	S	Reliability	Recognition	I	D	S	I	D	S	Reliability	Recognition
'0'	42	28	2	1	2	25	89.29	83.33	1	1	3	24	85.71	82.76					
'1'	42	28	0	1	1	26	92.86	92.86	0	1	2	25	89.29	89.29					
'2'	42	28	0	0	2	26	92.86	92.86	1	1	2	25	89.29	86.21					
'3'	42	28	0	0	0	28	100.00	100.00	0	0	1	27	96.43	96.43					
'4'	42	28	0	0	1	27	96.43	96.43	0	1	2	25	89.29	89.29					
'5'	42	28	0	1	1	26	92.86	92.86	0	1	1	26	92.86	92.86					
'6'	42	28	1	1	1	26	92.86	89.66	1	1	1	26	92.86	89.66					
'7'	42	28	0	0	0	28	100.00	100.00	0	0	1	27	96.43	96.43					
'8'	42	28	1	0	2	26	92.86	89.66	1	0	2	26	92.86	89.66					
'9'	42	28	0	1	0	27	96.43	96.43	0	1	1	26	92.86	92.86					
Total	420	280	4	5	10	265	94.64	93.31	4	7	16	257	91.79	90.49					

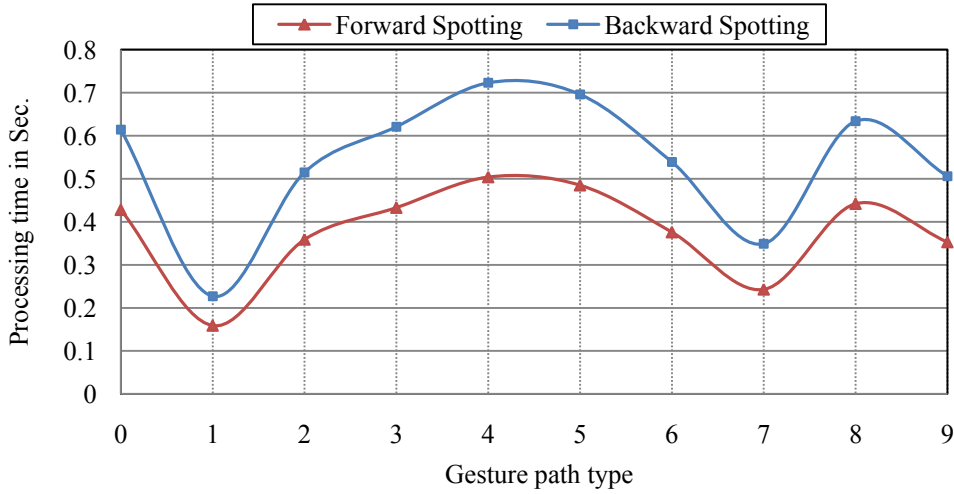


Figure 6.20: Average segmentation time of forward and backward spotting method.

long time than the forward spotting technique because the backward technique has to spend additional time for backtracking to find the gesture start point. In addition, when the average number of frames of a desired gesture increases, the difference in evaluation time between forward and backward spotting increases. The average number of frames for each gesture number is listed in Fig. B.1 in Appendix B.

It is shown that the proposed system has the ability to perform the hand gesture spotting and recognition tasks simultaneously for gesture numbers from 0 to 9. Furthermore, it is capable for real-time applications and solves the issues of time delay between the spotting and the recognition tasks.

6.5 Discussion and Conclusion

The main contribution of this chapter is to propose a forward gesture spotting method to simultaneously handles hand gesture spotting and recognition of numbers (0-9). To spot meaningful gestures of numbers accurately, a stochastic method was proposed for designing a non-gesture model with no training data. The non-gesture model with HMMs has been constructed by copying the states of all gesture models, each with an output observation probabilities. In CRFs, all the patterns other than gesture patterns are modeled by adding a label for non-gesture patterns. The non-gesture label is created using the weights of transition and state features function of initial CRFs. Moreover, the non-gesture model provides confidence measure which is used as an adaptive threshold to find the start and the end points of meaningful gestures.

The number of states for non-gesture model with HMMs increases as the number of gesture model increases. Furthermore, an increase in the number of states does not affect the recognition rate of the system and moreover it is a waste of time and space.

This problem was alleviated using relative entropy which merges similar probability distributions states. As a result, the number of states was decreased from 40 to 22 states, and in consequence, the model inference was faster and the evaluation time was saved $\approx 66.42\%$. On the other hand, it is difficult to spot and recognize short gestures with CRFs. It is because short gestures have fewer samples than long gestures. In order to avoid this problem, the weights of self-transition feature functions are increased. As a result, the reliability of CRFs method is improved from 86.12% to 90.49%

Another contribution was to use a forward spotting method. This method was based on two main modules: spotting module and recognition module. In spotting module, the sliding window was employed to calculate the observation probability of all gesture labels and non-gesture label (i.e. detect the start and the end points of meaningful gestures). The sliding window contains a number of sequential observations instead of a single observation. It is used to reduce the impact of observation changes for a short interval which are caused by incomplete feature extraction. The optimal value of sliding window is determined empirically with value 5 where the system shows the best performance in term of results. The gesture recognition module is activated after detecting the start point from continuous image sequences. The main objective is to perform the recognition process accumulatively for the segmented parts until it receives the end signal of key gestures and at this point, the observed gesture is recognized. Moreover, this method has solved the issues of time delay between the spotting and the recognition task. Experimental results show that the proposed system successfully spots and recognizes meaningful gestures with 93.31% and 90.49% reliability for HMMs and CRFs, respectively.

Chapter 7

Conclusions and Future Work

7.1 Thesis Summary

This dissertation investigated the problem of spotting and recognition of meaningful gestures which are embedded in the input video stream. One of such problems which arise in hand gesture recognition is to spot meaningful gestures from the continuous sequence of hand motions. Another problem is due to the variability in the same gesture even for the same person. Most of the approaches have used the backward spotting technique which causes inescapable time delay between the meaningful gesture spotting and recognition tasks.

The aim of the work was to propose a forward gesture spotting system to handle hand gesture segmentation and recognition at the same time. This system modeled gesture patterns discriminately and non-gesture patterns effectively. In addition, a stochastic method for designing a non-gesture model was proposed using HMMs versus CRFs models with no training data for non-gesture patterns. The non-gesture model provided a confidence measure which has been used as an adaptive threshold to find the start and the end points of meaningful gestures. Furthermore, the issues of time delay between the spotting and the recognition task has been solved.

The main findings of the thesis are summarized one by one in a sequel. Firstly, the fundamental techniques which build the basis for understanding this thesis have been briefly discussed. Different color models were explained; and after that, segmentation technique was exploited to segment hands and face which are biased to parametric modeling technique (e.g. Normal Gaussian distribution and Gaussian Mixture Models). A robust method for hand tracking in a complex environment using mean-shift algorithm in conjunction with depth map has been proposed. This structure correctly extracted a set of hand postures to track the hand motion and achieved accurate and robust hand tracking. Mean-shift analysis used the gradient of Bhattacharyya coefficient as a similarity function to derive the candidate of the hand which is most similar to a given hand target model. Depth information not only narrow down the search for objects of interest but it also increases the processing speed. Furthermore,

the depth information were used to completely solve complex background problem (i.e. neutralize complex background), as well as illumination variation. In case of the ambiguities (i.e. overlapping) between the hands and face, the depth information has successfully identified the objects under occlusion. Moreover, the optimization technique for mean-shift iteration reduced the computational time ≈ 20 times, which in turn made the system capable to real-time application.

A database contains 2440 video samples for gesture symbols where it captured by three persons on a set of twenty six alphabets and ten numbers. The input images were captured by Bumblebee stereo camera system which has 6 mm focal length at 15FPS with 240×320 pixels image resolution, Matlab implementation. Bumblebee stereo camera was used for acquisition of 2D images along with depth map. The experiments were carried out for an isolated gesture recognition and meaningful gesture spotting test. The isolated gestures have been handled according to two different classification techniques: a generative model such as HMMs and discriminative models like CRFs, HCRFs and LDCRFs. One HMM was constructed per gesture (i.e. each alphabet or number). Whereas, CRFs have been built using a single model for all reference gestures (i.e. one model for all alphabets and numbers). So, there is a trade-off for each gesture according to the weights of feature function. The HMMs parameters (i.e. the emission probability and the state transition matrix) have been learned from the same training data for CRFs. The HMMs were trained by BW algorithm while the CRFs were trained using gradient ascent with BFGS optimization technique. Training process was more expensive for CRFs than HMMs on a standard desktop PC since the time which CRFs need ranges from 20 minutes to several hours based on observation window. On the contrary, the recognition process is less costly and very fast for all models with sequences of several frames (i.e. requires a few seconds to recognize the sequence of frames).

One of main contribution using HMMs was to examine the capabilities of combined features of location, orientation and velocity for gesture recognition with respect to Cartesian and Polar coordinates. k -means clustering algorithm quantized the extracted features and employed them for the HMMs and CRFs codewords. It is noted that the effectiveness of these features yields reasonable recognition rates for alphabets and numbers. The results showed that the proposed system successfully recognizes isolated hand gestures with 94.75% recognition rate using $(Lc, Lsc, \theta_1, \theta_2, \theta_3, V)$ features. In addition, there was no large gap between LRB and LR topologies in term of results but the results of Ergodic topology were not promising when compared to LRB and LR topologies. On the other hand, LRB achieved promising results, and in consequence, it is employed as a basic model to carry out the recognition task. For discriminative models, CRFs, HCRFs and LDCRFs with different numbers of window size ranging from 0 to 7 have been applied and tested to decide the best in terms of their impact on gesture recognition. It is concluded that the optimal window size = 4 set empirically, when multiple experiments have been conducted with a variety of

window size to conclude the optimal for the system outcomes. The proposed system has automatically recognized tested gestures with 87.19%, 92.44%, 96.14% for CRFs, HCRFs and LDCRFs, respectively.

In contrast to generative and discriminative models, HMMs was the best in terms of results than CRFs, HCRFs and LDCRFs at window size = 0. The improvement in performance of discriminative structure for trained data was increased when the window size increases. As a result, LDCRFs was higher than HMMs according to the training and the testing data set at window size equal to 4. Our results showed that the overall recognition rates were 91.51%, 95.22%, 96.91% and 97.99% for CRFs, HCRFs, HMMs and LDCRFs, respectively. It is noted that the proposed system achieved high recognition rate due to a high segmentation accuracy of hand through the use of depth information. In addition, a good election for the set of feature candidates which optimally discriminate among input patterns. Also, a carefully experimental based selection of initialization parameters for training process. Above all, HMMs, CRFs, HCRFs and LDCRFs classification techniques have the ability to efficiently alleviate spatio-temporal variabilities.

To spot meaningful gestures of numbers from 0 to 9 accurately, a stochastic method was proposed for designing a non-gesture model without any training data for non-gesture patterns. The non-gesture model provides confidence measures which are used as an adaptive threshold to select the desired gesture model or spotting meaningful gestures (i.e. find the start and the end points of meaningful gestures which are embedded in the input video stream). The start and the end points of gestures were based on the observation probability value which was determined by the difference of observation probability (DP value) of maximal gesture models and non-gesture model. The transition from non-gesture to gesture occurs when the DP value changes from negative to positive (i.e. meaningful gestures start). Similarly, the transition from gesture to non-gesture occurs at the time when the DP value changes from positive to negative (i.e. meaningful gestures end). These observations have been employed as a rule to detect the start and the end point of gestures. The number of states for non-gesture model with HMMs increases as the number of gesture model increases. Furthermore, an increase in the number of states does not affect the recognition rate of the system and moreover it is a waste of time and space. This problem was alleviated using relative entropy which merged similar probability distributions states. As a result, the number of states was decreased from 40 to 22 states, and in consequence, the model inference was faster and the evaluation time was saved $\approx 66.42\%$. The reliability of CRFs methods have been improved by increasing the weights of self-transition feature for a short gestures to deal efficiently with spatio-temporal variabilities. Thus, the system has been appropriated to real-time implementations.

Another contribution was to use a forward spotting method in conjunction with different size of sliding window ranging from 1 to 8. Forward spotting was based on

two main modules: spotting module and recognition module. In spotting module, the start and the end points of meaningful gestures have been detected by DP value. Moreover, the gesture recognition module has been activated after detecting the start point from continuous image sequences. The main objective was to perform the recognition process accumulatively for the segmented parts until it receives the end signal of meaningful gestures. Furthermore, the comparison results between HMMs and CRFs were in the best at sliding window size equal to 5 empirically. The results showed that the proposed system successfully spotted and recognized meaningful gestures which are embedded in the input video stream with 93.31% and 90.49% reliability for HMMs and CRFs, respectively.

7.2 Future Work

The main objective of the proposed system was to make gesture spotting and recognition beneficial in a wide range of applications which impose very few restrictions on the users. We think the proposed system represents an important step towards achieving that goal, however there is still much room for improvement. The future work will focus on improving hand localization which will lead to extract more meaningful hand features and matching score. Specially at this point will be to use Adaboost-based face detector that has a key role in locating the left and right hands with respect to the face and the body of the gesturer. Adaboost technique can also be employed in 3D gesture recognition as a good choice to judge the quality of different features sets, combine these sets and finally distill the optional subsets.

The representation of mean-shift algorithm has proven to be robust when dealing with changes in shape, size, partial occlusion as well as stop and go conditions. In order to consider the complete occlusion, an adaptive threshold can be assigned to the similarity measure, in addition to adaptively skip a number of frames. The risk here is to lose the true position of the target. Moreover, it is also possible to combine the mean-shift with Kalman filter as another solution to alleviate the occlusion problem. Here, the measurement vector is determined based on mean shifts and then the next hand location is predicated by Kalman filter to predict and correct the states of linear processes.

We also expect an ongoing increase of work on extending the proposed non-gesture model with CRFs to LDCRFs. This expectation is build on try-and-errors of the optimization parameters of LDCRFs model. In addition, the extended system can included the words and sentences of sign language along with facial features of the face. The objective behind the use of these multi-model system is to envision a wide range of real-time applications and addresses the realistic situation. Furthermore, the consideration of grammar is an important feature for continuous sign language recognition. Moreover, the grammar will predict the appearance of the sign in the context of the previously seen observations.

Each gesture can begin and end with a specific hand shape (i.e. posture). So, in order to detect and distinguish the gestures/postures with different hand shapes and trajectories, the integration of posture and gesture can be considered for the utterance data set. The motivation behind integration is to extract multiple signs at the same time driven from different approaches (i.e. from gesture and posture systems) and results in the inference of a new symbol.

The future research will also address the hand gestures by using fingertip trajectory instead of hand centroid points to spot and recognize them with multi-camera system. The aim is to consider both local and global hand motions, readiness of the system to work in the event of loss of depth information from any camera, and represents the hand gesture in small space which makes the system more realistic for the applications. In addition, the gesture can be divided into subunits to speedup the computational time for obtaining the best matching gestures. The main challenge here is the decision of the number of subunits and the models to employ for those subunits.

Appendices

Appendix A

Data Processing

This appendix explores three parts; skin and non-skin color database, clustering the features which are extracted from hand gesture path, and mean-shift procedure for the hand trajectory.

A.1 skin and non-skin database

In training data set, 18972 skin pixels from 36 different races persons and 88320 non-skin pixels from 84 different images are used. The skin images which contain human subjects have been downloaded from the internet. The database of skin images used

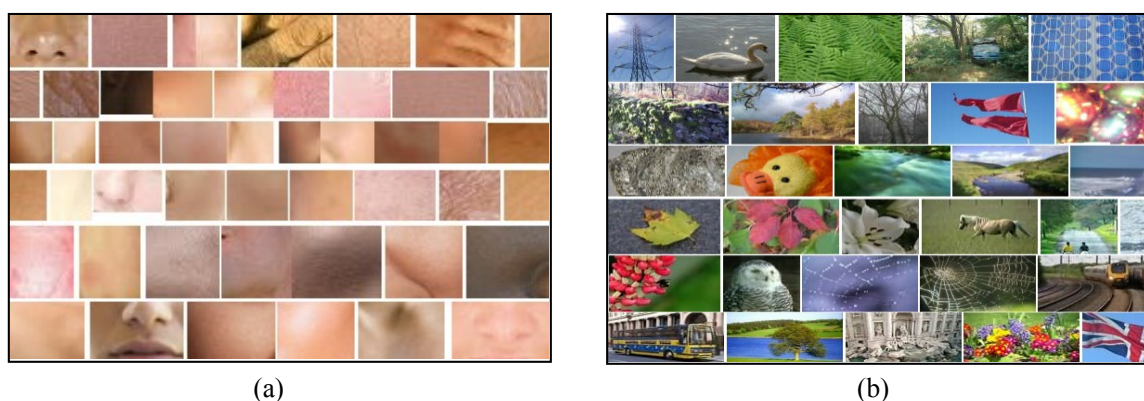


Figure A.1: Cropped images for skin and non-skin pixels that were collected from the World Wide Web. (a) Database of skin pixels for different races. (b) Database of non-skin pixels for different background.

in the work is shown in Fig. A.1(a). Similarly, the non-skin images (i.e. background regions) from the internet are collected and modeled for the system. The database of non-skin images is shown in Fig. A.1(b). As discussed in Section 4.1.1, the skin

color is localized in a small region of the chrominance (C_b, C_r) space as can be seen in Fig. A.2(a). So, GMMs technique begins with modeling of skin by using skin database where a variant of k -means clustering algorithm performs the model training to determine the mean vector, covariance matrix and mixture weight. The number of Gaussian components relies on the skin database used and is automatically estimated by a constructive algorithm [86] which uses the criteria of maximizing likelihood function.

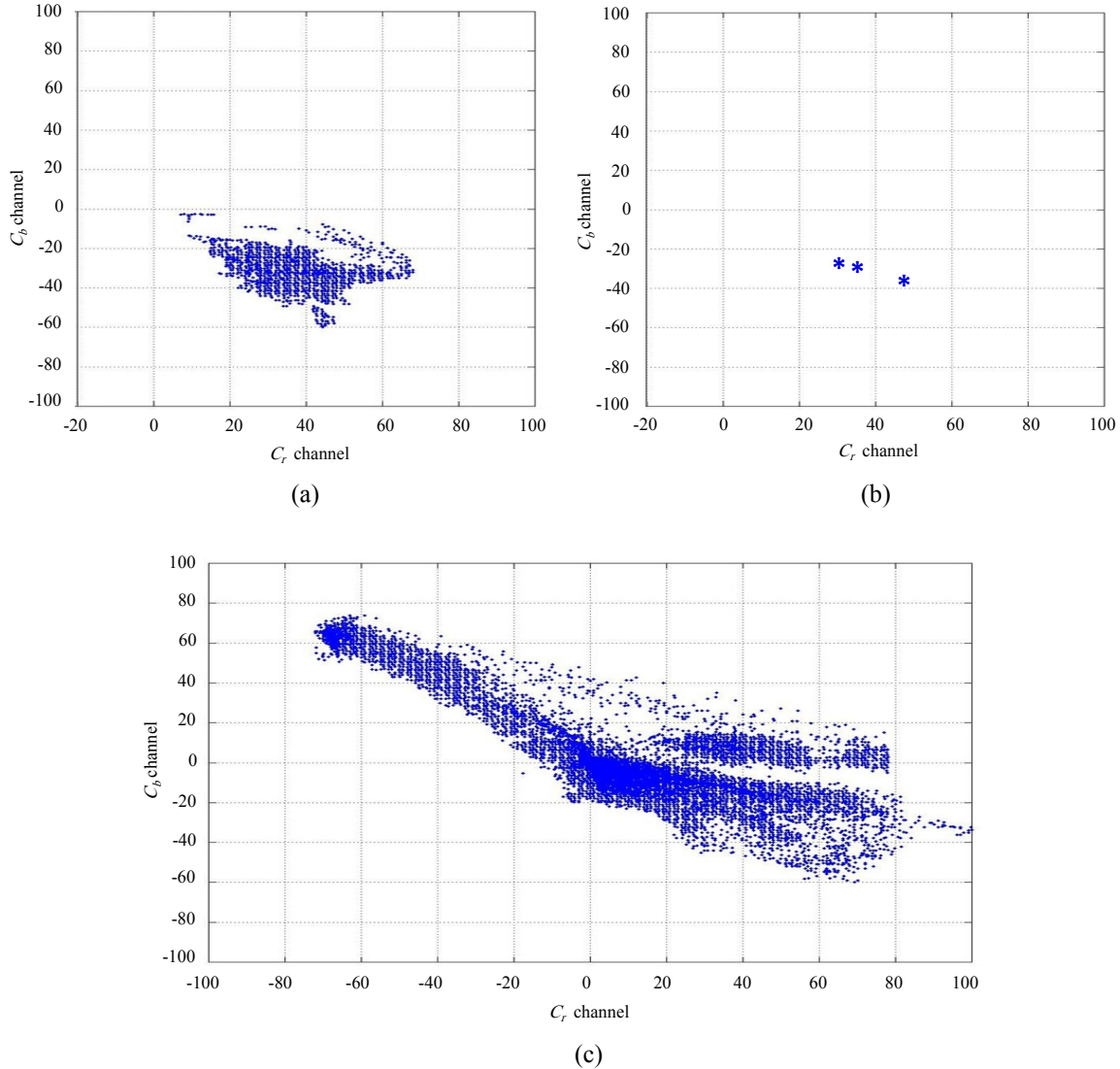


Figure A.2: Distribution values of skin and non-skin pixels projected onto the (C_b, C_r) plane for training data. (a) Distribution values of skin pixels for training data where the skin color is localized to a small region in the (C_b, C_r) chrominance space. (b) Location of the mean points according to three components of Gaussian Mixture Models for skin database. (c) Non-skin pixels distribution for training data.

A.2 Cluster Hand Trajectory

This part illustrates the cluster trajectories for gesture numbers (0-9) according to $(Lc, Lsc, \theta_1, \theta_2, \theta_3, V)$ features in Cartesian coordinate. In this manner, gesture is represented as an ordered sequence of feature vectors which are projected and clustered using k -means algorithm in space dimension to obtain discrete codewords that are used as an input to HMMs and CRFs [124, 126, 127].

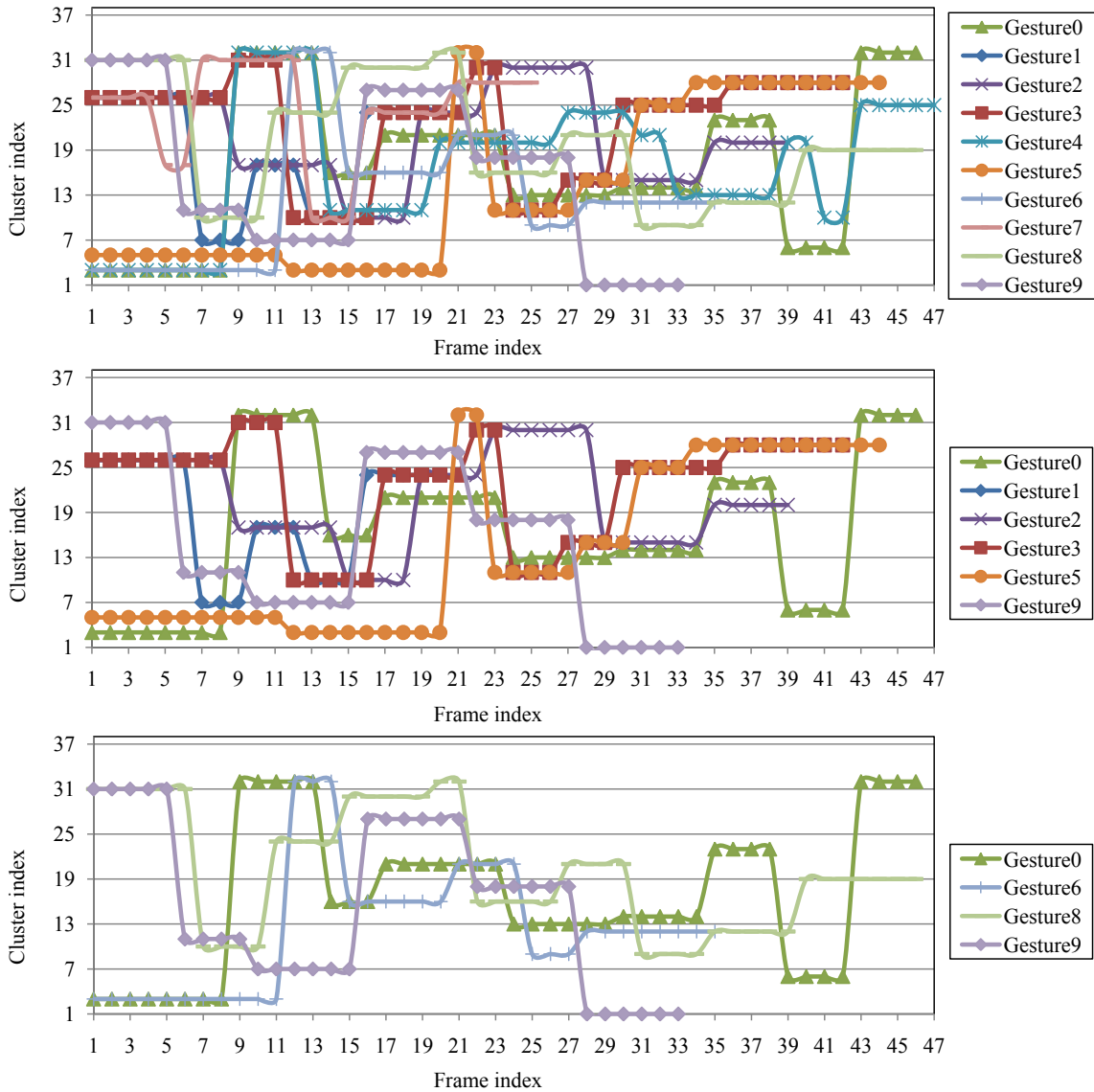


Figure A.3: Cluster trajectories in Cartesian system for gesture numbers according to $(Lc, Lsc, \theta_1, \theta_2, \theta_3, V)$ features. The middle and bottom graphs are the same of the top graph after eliminating the different cluster trajectories. Here, gesture paths '0' and '6' have the same cluster indices until frame 33.

Fig. A.4 shows the Cluster trajectories of gesture path ‘3’ and ‘5’, which are projected according to $(L_c, L_{sc}, \theta_1, \theta_2, \theta_3, V)$, (ρ_c, φ_c) and $(\theta_1, \theta_2, \theta_3)$ features, respectively. It is noted that the cluster trajectories for gesture paths ‘3’ and ‘5’ in the top graph nearly have the same cluster indices from frame 21 to frame 43. So, this proves the reality of combined features $(L_c, L_{sc}, \theta_1, \theta_2, \theta_3, V)$ in Cartesian coordinate.

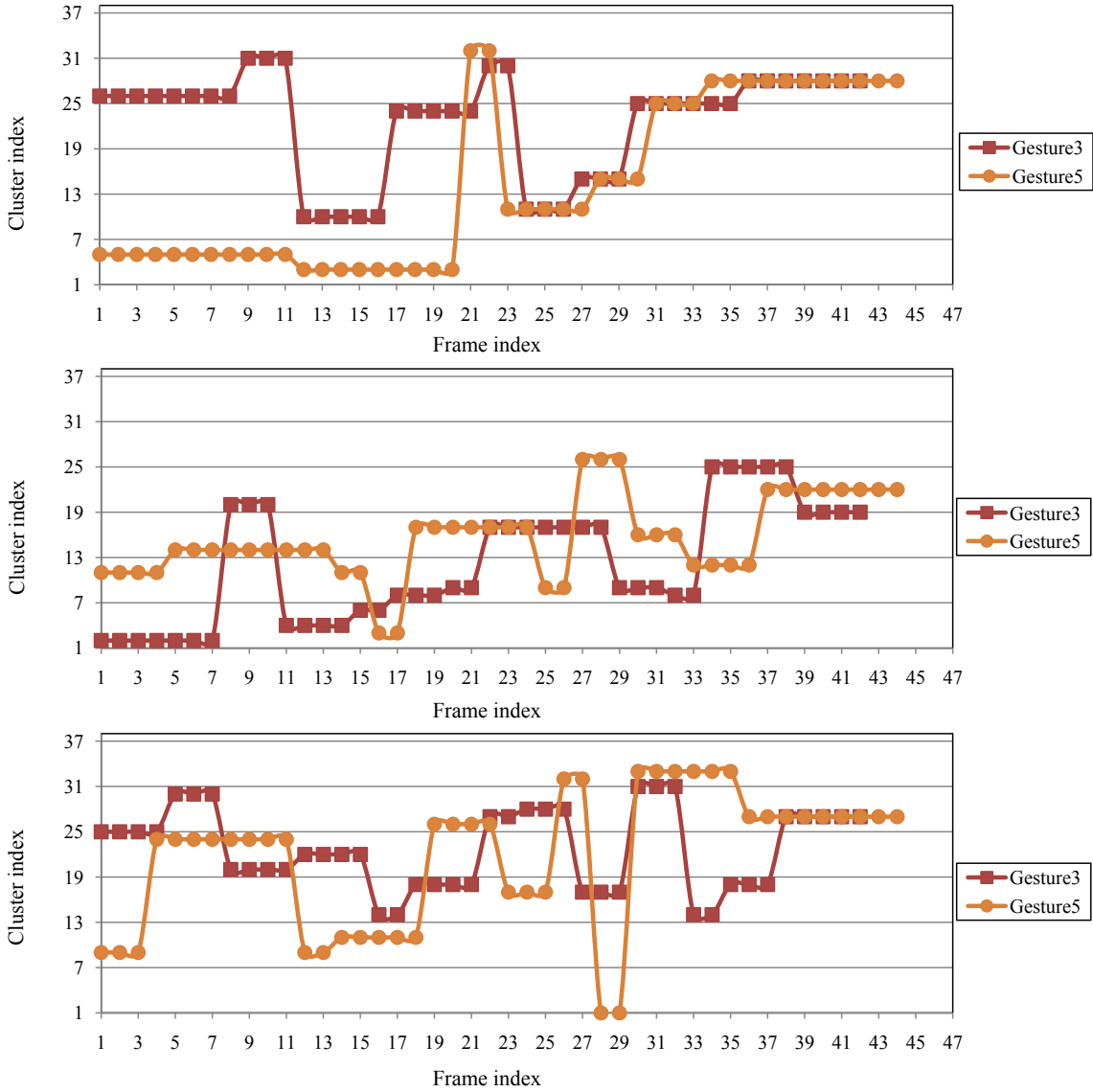


Figure A.4: Cluster trajectories for gesture ‘3’ and gesture ‘5’ according to features $(L_c, L_{sc}, \theta_1, \theta_2, \theta_3, V)$, (ρ_c, φ_c) and $(\theta_1, \theta_2, \theta_3)$, respectively. The cluster trajectories which are depicted in the middle and bottom graphs are varying than the top graph, notably in the later parts of gesture paths ‘3’ and ‘5’.

A gesture is spatio-temporal pattern which may be static, dynamic or both. So, there is a quite bit of variability (i.e. in shape, trajectory and duration) in the same gesture even for the same person. The following figure illustrates varying trajectories of gesture ‘3’ for the same person. The cluster trajectories of these gestures have the same cluster indices but with slight variations.

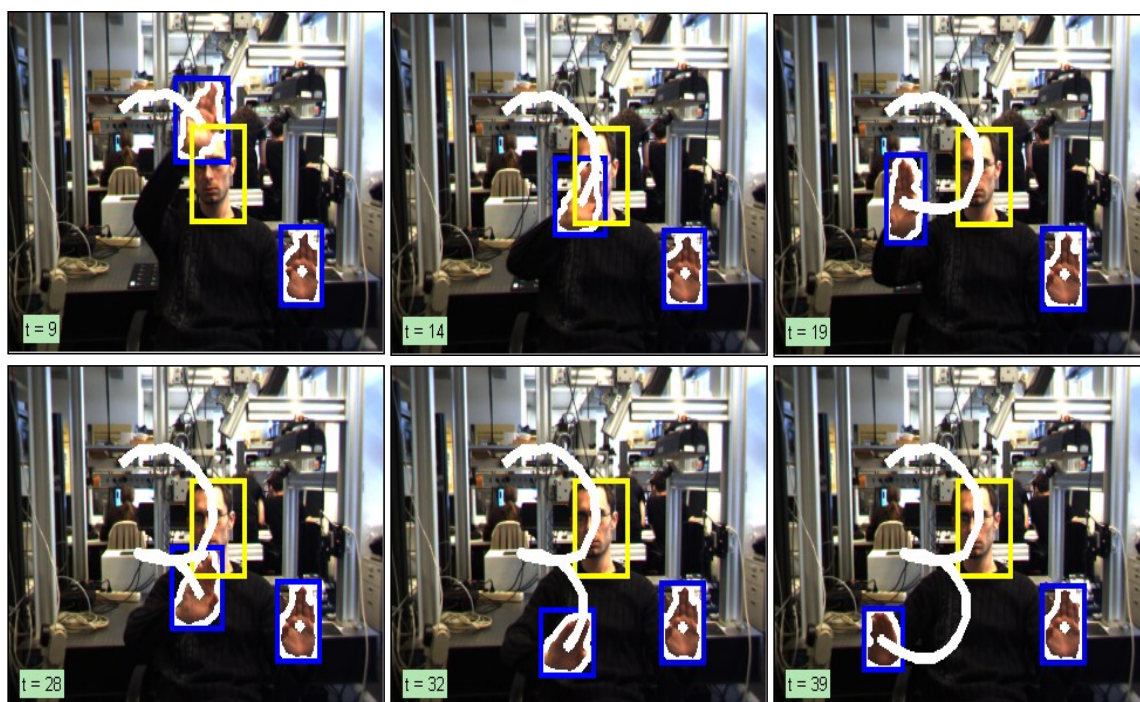
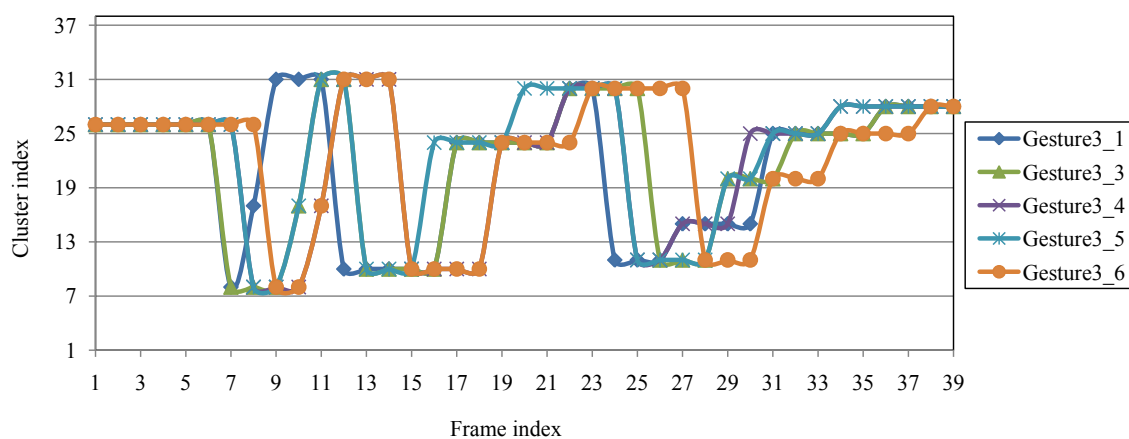


Figure A.5: Cluster trajectories for the gesture path ‘3’ with respect to different five video samples. It is noted that the same gesture have similar cluster indices but with slight variations in their cluster trajectories (i.e. spatio-temporal variabilities).

A.3 Mean-shift Analysis

According to Section 4.2.1, mean-shift iteration uses the gradient of Bhattacharyya coefficient as a similarity function to indicate the direction of hand's movement. Moreover, the mean-shift procedure is defined recursively and performs the optimization to compute the mean-shift vector.

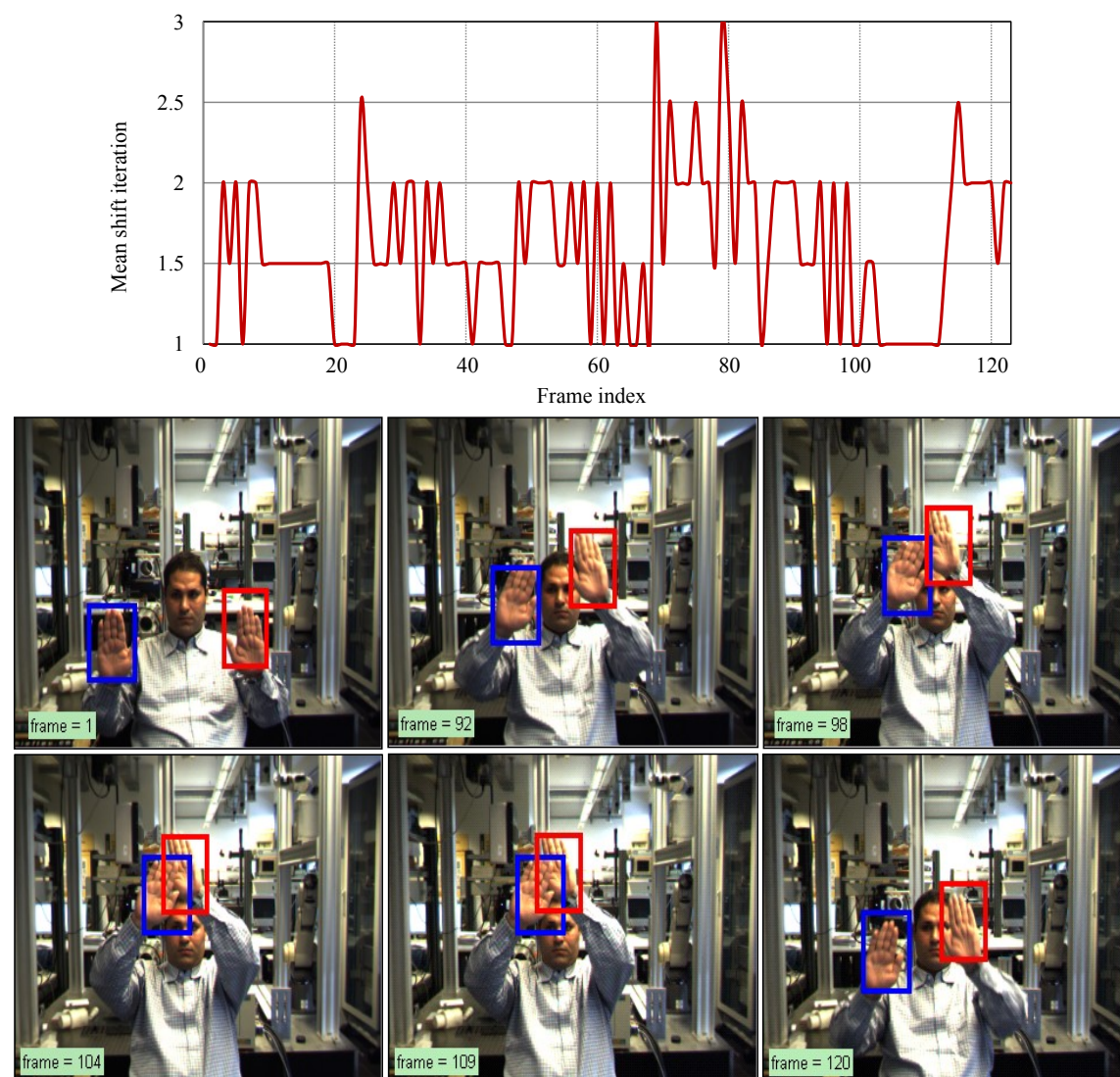


Figure A.6: Tracking result where at frame 109, both hands are correctly determined notably in case of overlapping and partial occlusion. In top figure, the number of mean-shift iteration is 1.61 per frame for both left and right hands, which in turn makes the system capable for real-time implementation.

Appendix B

Classification Results

This appendix explores two parts. The first part is related to some results of isolated gestures using HMMs and LDCRFs while the second part shows some results of spotting meaningful gestures.

B.1 Isolated gestures











Sign	Segmeted parts	Average length	Recognition by HMMs	Problem
0		46	93.33%	Confusion with G and 6
1		19	100.00%	
2		39	96.67%	Confusion with Z
3		42	100.00%	
4		47	100.00%	
5		44	93.33%	Confusion with S
6		35	96.67%	Confusion with 0
7		25	100.00%	
8		46	96.67%	Confusion with 3
9		34	100.00%	

Figure B.1: Hand gesture paths for gesture numbers from 0 to 9 with segmented parts.


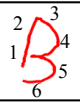

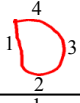
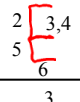





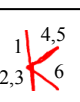


Sign	Segmeted parts	Average length	Recognition by HMMs	Problem
A		39	100.00%	
B		58	100.00%	
C		25	93.33%	Confusion with G and 6
D		44	96.67%	Confusion with 0
E		41	100.00%	
F		20	100.00%	
G		31	93.33%	Confusion with C and 6
H		34	96.67%	Condusion with K
I		11	100.00%	
J		20	100.00%	
K		64	96.67%	Confusion with H
L		18	100.00%	
M		48	100.00%	

Figure B.2: Hand gesture paths for alphabets from A to M with segmented parts.














Sign	Segmeted parts	Average length	Recognition by HMMs	Problem
N		35	100.00%	
O		36	100.00%	
P		26	96.67%	Confusion with B
Q		26	100.00%	
R		60	96.67%	Confusion with B
S		39	93.33%	Confusion with 5
T		39	100.00%	
U		37	100.00%	
V		25	96.67%	Confusion with W
W		38	96.67%	Confusion with U
X		35	100.00%	
Y		32	100.00%	
Z		32	96.67%	Confusion with 2

Figure B.3: Hand gesture paths for alphabets from N to Z with segmented parts.

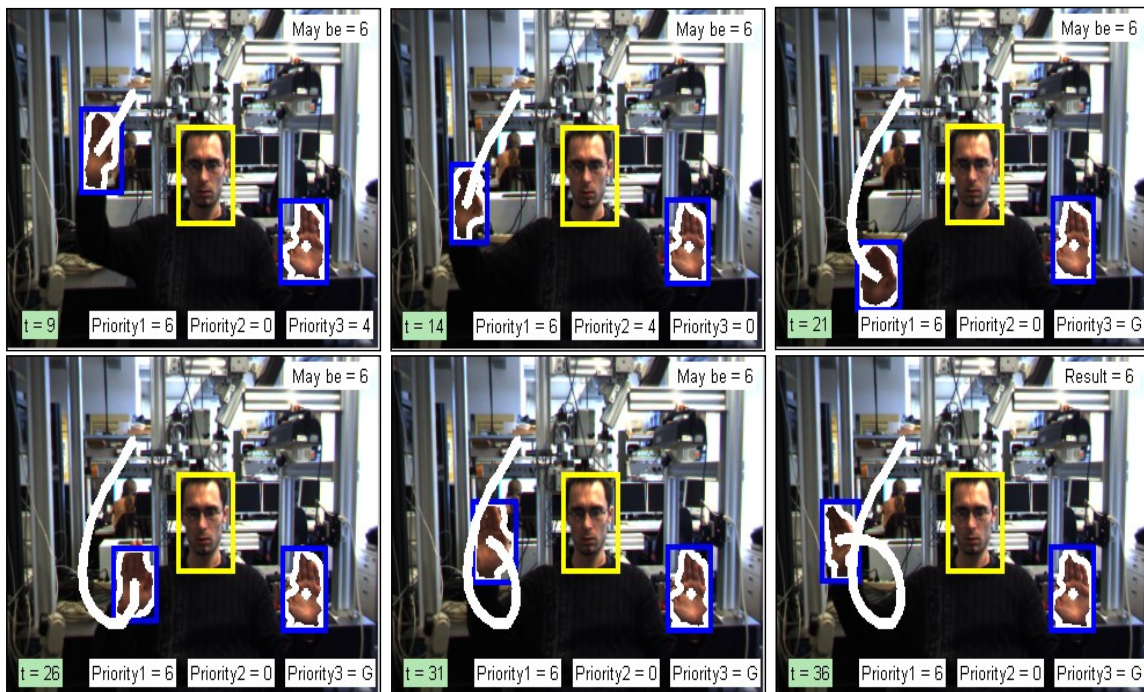
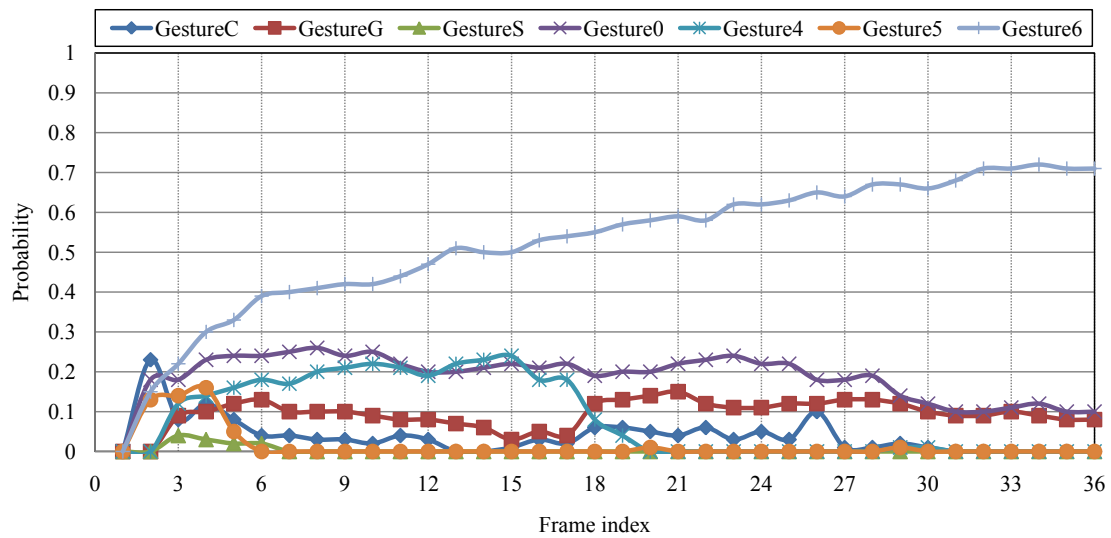


Figure B.4: Temporal evolution of the seven higher probabilities of the gestures ‘C’, ‘G’, ‘S’, ‘0’, ‘4’, ‘5’ and ‘6’ using LDCRFs. In the image sequences, the highest priority is gesture number ‘6’ at frame 21 as well as in frame 31, and at frame 36 the result is gesture number ‘6’.

B.2 Gesture spotting

Forward spotting is based on two main modules; spotting module and recognition module. In spotting module, the sliding window is employed to detect the start and the end points of meaningful gestures. The gesture recognition module is fired after detecting the start point to accumulatively perform the recognition process until receiving the end signal of meaningful gesture.

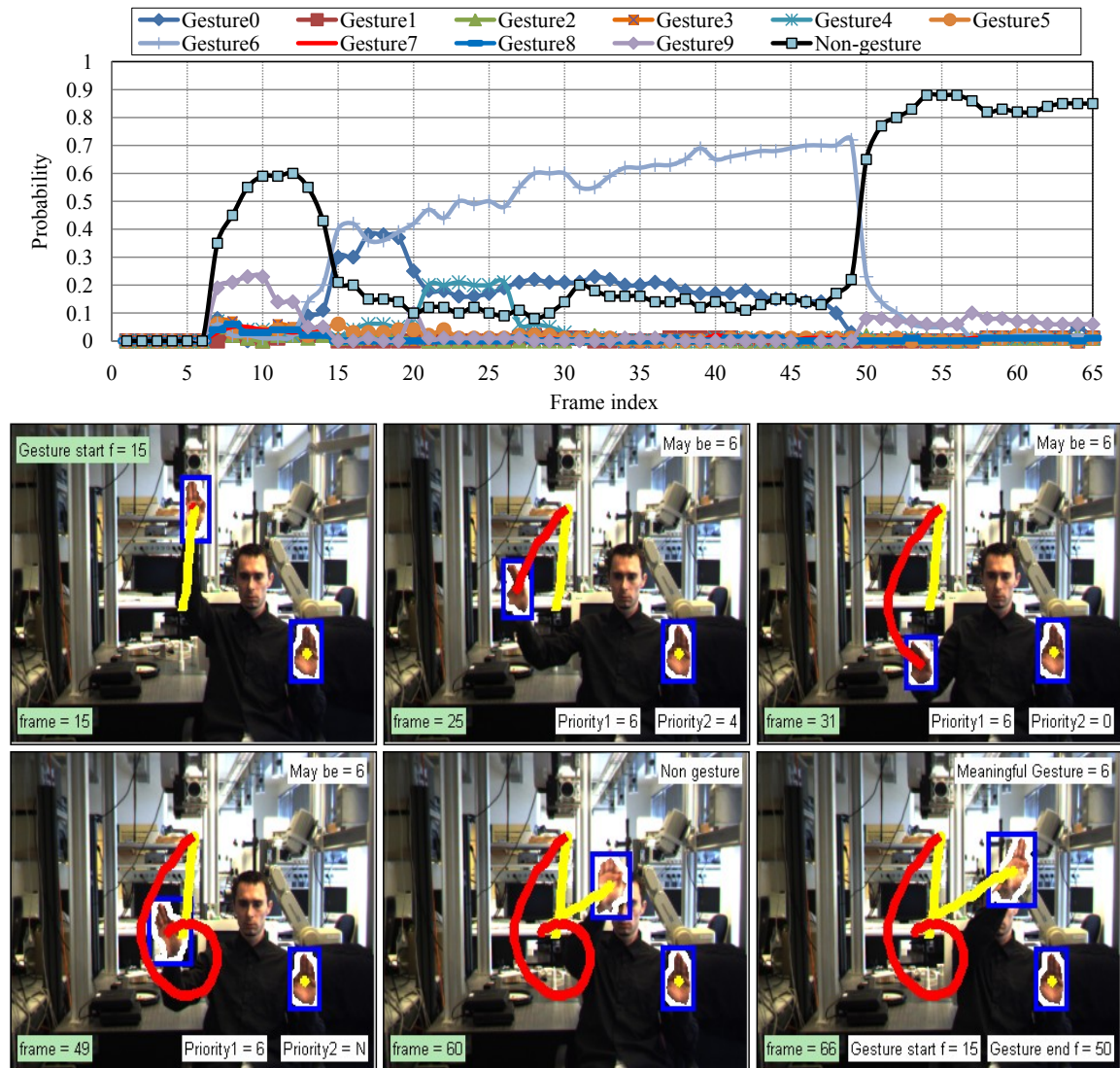


Figure B.5: Temporal evolution of the probabilities of the gestures number (0-9) and non-gesture label 'N'. The image sequences contain one meaningful gestures '6'. At frame 15, the start point is detected since the highest priority is assigned to gesture labels than the non-gesture label. At frame 50, the highest priority is assigned to non-gesture label which means that the end point of meaningful gesture '6' is detected.

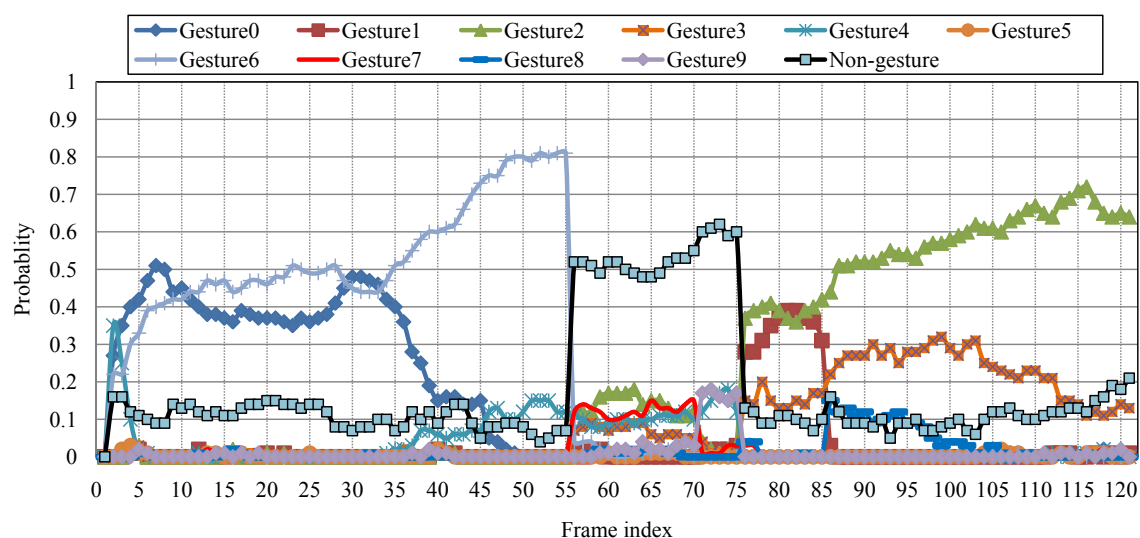


Figure B.6: Temporal evolution of the probabilities of the gestures number (0-9) and non-gesture label 'N'. The image sequences contain two key gestures '6' '2', where the end point of gesture '6' is at frame 56 and the start point of gesture '2' is at frame 76. In the first 55 frames, the probability of non-gesture label is not the maximum value, which means that the end point of the key gesture is not detected. At frame 56, the first key gesture '6' ends where the non-gesture label has a high probability than other gesture labels. Between frame 56 and frame 75, the highest priority is assigned to non-gesture label, which means that the start point of the second key gesture is not detected. At frame 76, a new key gesture is started, where the probability value of non-gesture label is not the highest value as compared to the other gesture labels.

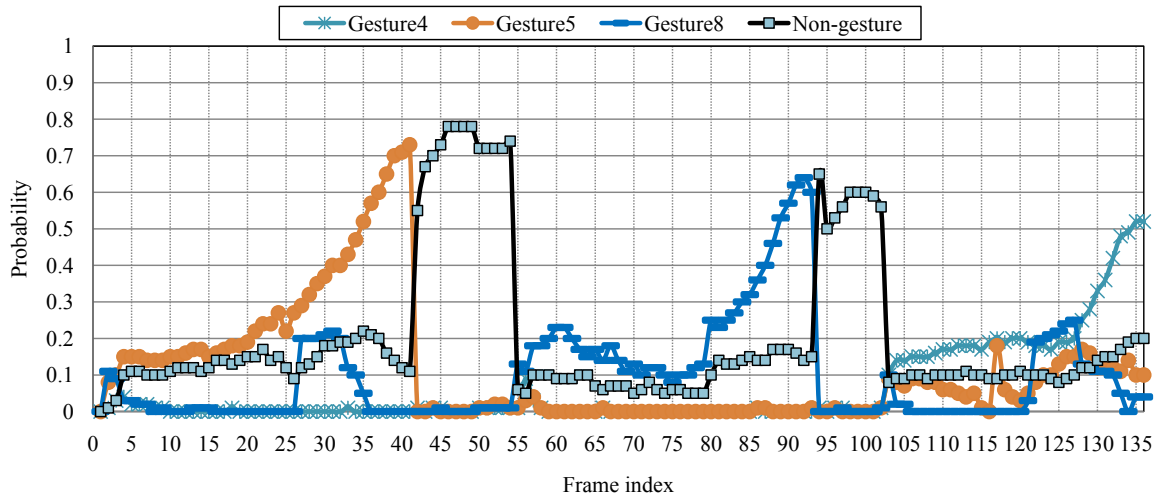


Figure B.7: Temporal evolution of the probabilities of the gestures number ‘4’, ‘5’, ‘8’ and non-gesture label ‘N’. The image sequences contain three key gestures ‘5’, ‘8’, ‘4’. The end point of gesture ‘5’ is at frame 41. Between frame 42 and frame 56, the highest priority is assigned to non-gesture label, which means that the start point of the second key gesture is not detected. At frame 57, a new key gesture is started where the probability value of non-gesture label is not the highest value as compared to the other gesture labels. The end point of gesture ‘8’ is at frame 93. Between frame 94 and frame 102, the highest priority is assign to non-gesture label. The start point of gesture path ‘4’ is at frame 103. The final result of the continuous gesture path is ‘584’.

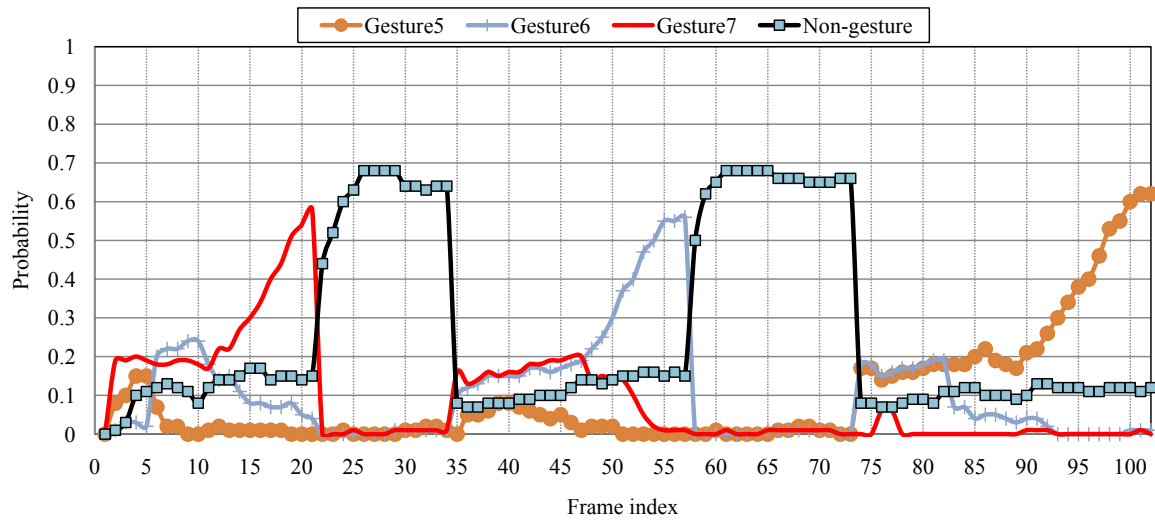


Figure B.8: Temporal evolution of the probabilities of the gestures number ‘5’, ‘6’, ‘7’ and non-gesture label ‘N’. The image sequences contain three key gestures ‘7’, ‘6’, ‘5’. The end point of gesture ‘7’ is at frame 21. Between frame 22 and frame 34, the highest priority is assigned to non-gesture label, which means that the start point of the second key gesture is not detected. At frame 35, a new key gesture is started where the probability value of non-gesture label is not the highest value as compared to the other gesture labels. The end point of gesture ‘6’ is at frame 57. Between frame 58 and frame 73, the highest priority is assign to non-gesture label. The start point of gesture path ‘5’ is at frame 74. The final result of the continuous gesture path is ‘765’.

Bibliography

- [1] N. P. Vassilia and G. M. Konstantinos, “On Feature Extraction and Sign Recognition for Greek Sign Language,” *International Conference on Artificial Intelligence and Soft Computer*, pp. 93–98, 2003.
- [2] F. Quek, “Toward a Vision-based Hand Gesture Interface,” *Proceedings of Virtual Reality Software and Technology Conference*, pp. 17–31, 1994.
- [3] T. F. E. Wikipedia, “http://en.wikipedia.org/wiki/K-means_clustering.”
- [4] Y. Wu and T. S. Huang, “Hand Modeling Analysis and Recognition for Vision-based Human Computer Interaction,” *IEEE Signal Processing Magazine, Special Issue on Immersive Interactive Technology, Vol. 18, No. 3*, pp. 51–60, 2001.
- [5] T. Starner, J. Weaver, and A. Pentland, “Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video,” *IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 20, No. 12*, pp. 1371–1375, 1998.
- [6] H. Yang, A. Park, and S. Lee, “Gesture Spotting and Recognition for Human-Robot Interaction,” *IEEE Transaction on Robotics, Vol. 23, No. 2*, pp. 256–270, 2007.
- [7] W. Freeman and M. Roth, “Orientation Histograms for Hand Gesture Recognition,” *In International Workshop on Automatic Face and Gesture Recognition*, pp. 296–301, 1995.
- [8] S. Ju, M. Black, S. Minneman, and D. Kimber, “Analysis of Gesture and Action in Technical Talks for Video Indexing,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 595–601, 1997.
- [9] V. Nair and J. J. Clark, “Automated Visual Surveillance Using Hidden Markov Models,” *Conference of Vision Interface*, pp. 88–92, 2002.
- [10] R. C. Rose, “Discriminant Wordspotting Techniques for Rejection Non-vocabulary utterances in Unconstrained Speech,” *IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2*, pp. 105–108, 1992.

-
- [11] F. R. Chen, L. D. Wilcox, and D. S. Bloomberg, "Word Spotting in Scanned Images using Hidden Markov Models," *IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 5*, pp. 1–4, 1993.
- [12] K. Takahashi, S. Seki, , and R. Oka, "Spotting Recognition of Human Gestures from Motion Images," *Technical Report IE92-134*, pp. 9–16, 1992.
- [13] T. Baudel and M. Beaudouin, "CHARADE: Remote Control of Objects using Free-Hand Gestures," *Communications of ACM, Vol. 36, No. 7*, pp. 28–35, 1993.
- [14] A. Wexelblat, "Natural Gesture in Virtual Environments," *Proceedings of Virtual Reality Software and Technology Conference*, pp. 5–16, 1994.
- [15] H. Lee and J. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 21, No. 10*, pp. 961–973, 1999.
- [16] H. Yang, S. Sclaroff, and S. Lee, "Sign Language Spotting with a Threshold Model Based on Conditional Random Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 7*, pp. 1264–1277, 2009.
- [17] J. Alon, V. Athitsos, Y. Quan, and S. Sclaroff, "A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 9*, pp. 1685–1699, 2009.
- [18] X. Deyou, "A Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG," *In International Conference on Pattern Recognition*, pp. 519–522, 2006.
- [19] L. Rabiner and B. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, pp. 4–16, 1996.
- [20] C. Keskin, O. Aran, and L. Akarun, "Real Time Gestural Interface for Generic Applications," *European Signal Processing Conference, EUSIPCO Demonstration Session, Antalya, 2005*.
- [21] C. Keskin, K. Balci, O. Aran, B. Sankur, and L. Akarun, "A Multimodal 3D Healthcare Communication System," *In: 3DTV Conf., Kos., 2007*.
- [22] C. Vogler and D. Metaxas, "A Framework for Recognizing the Simultaneous Aspects of American Sign Language," *Journal of Computer Vision and Image Understanding, Vol. 81, No. 3*, pp. 358–384, 2001.

- [23] ———, “Handshapes and Movements: Multiple-channel American Sign Language Recognition,” *Lecture Notes in Computer Science, Springer Berlin/Heidelberg, ISBN: 978-3-540-21072-6*, pp. 431–432, 2004.
- [24] N. Tanibata, N. Shimada, and Y. Shirai, “Extraction of Hand Features for Recognition of Sign Language Words,” *International Conference on Vision Interface*, pp. 391–398, 2002.
- [25] A. McCallum, D. Freitag, and F. Pereira, “Maximum Entropy Markov Models for Information Extraction and Segmentation,” *International Conference on Machine Learning*, pp. 591–598, 2000.
- [26] J. Lafferty, A. McCallum, and F. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling sequence Data,” *International Conference on ICML*, pp. 282–289, 2001.
- [27] S. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell, “Hidden Conditional Random Fields for Gesture Recognition,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1521–1527, 2006.
- [28] P. Morguet and M. Lang, “Spotting Dynamic Hand Gestures in Video Image Sequences Using Hidden Markov Models,” *IEEE International Conference on Image Processing*, pp. 193–197, 1998.
- [29] R. Oka, “Spotting Method for Classification of Real World Data,” *The Computer Journal, Vol. 41, No. 8*, pp. 559–565, 1998.
- [30] Y. Ho-Sub, S. Jung, J. B. Young, , and S. Y. Hyun, “Hand Gesture Recognition using Combined Features of Location, Angle and Velocity,” *Journal of Pattern Recognition, Vol. 34, No. 7*, pp. 1491–1501, 2001.
- [31] Y. Zhu, G. Xu, and D. Kriegman, “A Real-time Approach to the Spotting, Representation, and Recognition of Hand Gestures for Human-computer Interaction,” *Journal of Computer Vision and Image Understanding, Vol. 85, No. 3*, pp. 189–208, 2002.
- [32] H. Kang, C. Lee, and K. Jung, “Recognition-based Gesture Spotting in Video Games,” *Journal on Pattern Recognition Letters, Vol. 25, No. 15*, pp. 1701–1714, 2004.
- [33] K. Kahol, P. Tripath, and S. Panchanathan, “Automated Gesture Segmentation from Dance Sequences,” *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 883–888, 2004.

- [34] J. Alon, V. Athitsos, and S. Scharoff, "Accurate and Efficient Gesture Spotting via Pruning and Sungesture Reasoning," *IEEE ICCV Workshop on Human Computer Interaction, Lecture Note in Computer Science 3766*, pp. 189–198, 2005.
- [35] T. Darrell, I. Essa, and A. Pentland, "Task-specific Gesture Analysis in Real-time Using Interpolated Views," *Journal of Pattern Analysis and Machine Intelligence, Vol. 18, No. 12*, pp. 1236–1242, 1996.
- [36] J. B. Kruskal and M. Liberman, "The symmetric Time Warping Algorithm: From Continuous to Discrete," *In Time Warps, String Edits and Macromolecules, J. B. Kruskal and D. Sankoff, Eds. Addison-Wesley*, pp. 125–162, 1993.
- [37] N. Stefanov, A. Galata, and R. Hubbard, "Real-time Hand Tracking with Variable-length Markov Models of Behaviour," *In Real Time Vision for Human-Computer Interaction*, pp. III:73–73, 2005.
- [38] F. Chen, C. Fu, and C. Huang, "Hand Gesture Recognition Using a Real-time Tracking Method and Hidden Markov Models," *Journal of Image and Video Computing, Vol. 21, No. 8*, pp. 745–758, 2003.
- [39] A. Quattoni, S. Wang, L. P. Morency, M. Collins, and T. Darrell, "Hidden Conditional Random Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 10*, pp. 1848–1852, 2007.
- [40] A. P. H. Yang and S. Lee, "Robust Spotting of Key Gestures from Whole Body Motion Sequence," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 231–236, 2006.
- [41] W. C. S. Jr., "Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf," *J. Deaf Stud. Deaf Educ., Vol. 10, No. 1*, pp. 3–37, 2005.
- [42] H. Hienz, B. Bauer, and K. F. Kraiss, "HMM-based Continuous Sign Language Recognition Using Stochastic Grammars," *International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, pp. 185–196, 1999.
- [43] B. Bauer and K. Kraiss, "Video-Based Sign Recognition Using Self-Organizing Subunits," *International Conference on Pattern Recognition*, pp. 434–437, 2002.
- [44] S. Akyol, "Nicht-intrusive Erkennung isolierter Gesten und Gebärden," *PhD Thesis, Aachen, Techn. Hochsch*, 2003.

- [45] J. Zieren and K. F. Kraiss, "Robust Person-Independent Visual Sign Language Recognition," *In IbPRIA*, pp. 520–528, 2005.
- [46] L. Nianjun, C. L. Brian, J. K. Peter, and A. D. Richard, "Model Structure Selection & Training Algorithms for a HMM Gesture Recognition System," *International Workshop in Frontiers of Handwriting Recognition*, pp. 100–106, 2004.
- [47] D. B. Nguyen, S. Enokida, and E. Toshiaki, "Real-Time Hand Tracking and Gesture Recognition System," *IGVIP Conference, CICC*, pp. 362–368, 2005.
- [48] 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*, Vol. 24, No. 8, pp. 1061–1074, 2002.
- [49] R. Stiefelhagen, C. Fügen, P. Giesemann, H. Holzapfel, K. Nickel, and A. Waibel, "Natural Human-robot Interaction Using Speech, Gaze and Gestures," *International Conference on Intelligent Robots and Systems*, pp. 2422–2427, 2004.
- [50] R. Kjeldsen and J. kender, "Visual Hand Gesture Recognition for Window System Control," *International Workshop on Face and Gesture Recognition*, pp. 184–188, 1995.
- [51] C. Maggioni, "Novel Gestural Input Device for Virtual Reality," *IEEE Virtual Reality Annual International Symposium*, pp. 118–124, 1993.
- [52] J. Alon, "Spatiotemporal Gesture Segmentation," *PhD thesis, Computer Science Dept., Boston Univ.*, 2006.
- [53] J. Jelinek, "Statistical Methods for Speech Recognition," *MIT Press*, 1997.
- [54] W. Gao, G. Fang, D. Zhao, and Y. Chen, "Transition Movement Models for Large Vocabulary Continuous Sign Language Recognition," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 553–558, 2004.
- [55] M. Szummer, "Learning Diagram Parts with Hidden Random Fields," *Conference on Document Analysis and Recognition*, pp. 1188–1193, 2005.
- [56] K. Vaananen and K. Boehm, "Gesture Driven Interaction as a Human Factor in Virtual Environments-An Approach with Neural Networks," *Virtual Reality Systems, R. Earnshaw, M. Gigante, H. Jones, eds., chapter 7, Academic Press*, pp. 93–106, 1993.
- [57] S. Fels, "Glove-Talk II: Mapping Hand Gestures to Speech Using Neural NetworksAn Approach to Building Adaptive Interfaces," *Ph.D. Thesis, University of Toronto*, 1994.

- [58] S. Waldherr, R. Romero, and S. Thrun, "A Gesture Based Interface for Human-Robot Interaction," *Journal of Autonomous Robots*, Vol. 9, No. 2, pp. 151–173, 2000.
- [59] D. Kortenkamp, E. Huber, and R. P. Bonasso, "Recognizing and Interpreting Gestures on a Mobile Robot," *In Proceedings of AAAI-96*, pp. 915–921, 1996.
- [60] S. Seki, K. Takahashi, and R. Oka, "Gesture Recognition from Motion Images by Spotting Algorithm," *In Proceedings of Asia Conference on Computer Vision*, pp. 759–762, 1993.
- [61] J. Alon, V. Athitsos, Y. Quan, and S. Sclaroff, "Simultaneous Localization and Recognition of Dynamic Hand Gestures," *IEEE Workshop Motion and Video Computing*, pp. 254–260, 2005.
- [62] R. Oka, "Spotting Method for Classification of Real World Data," *The Computer Journal*, Vol. 41, No. 8, pp. 559–565, 1998.
- [63] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, 1989.
- [64] X. Kahol, "Gesture Segmentation in Complex Motion Sequences," *Masters thesis, Arizona State University, Tempe, AZ*, 2003.
- [65] A. Braffort, "ARGo: An Architecture for Sign Language Recognition and Interpretation," *International Gesture Workshop Progress in Gestural Interaction*, pp. 17–30, 1996.
- [66] K. Kahol, P. Tripath, and S. Panchanthan, "Documenting Motion Sequences: Development of a Personalized Annotation System," *IEEE Multimedia Magazine*, Vol. 13, No. 1, pp. 35–47, 2006.
- [67] D. Pinto, A. McCallum, X. Wei, and W. B. Croft, "Table Extraction Using Conditional Random Fields," *Proceedings of the ACM SIGIR*, 2003.
- [68] F. Sha and F. Pereira, "Shallow Parsing with Conditional Random Fields," *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 134–141, 2003.
- [69] R. Yang and S. Sarkar, "Detecting Coarticulation in Sign Language Using Conditional Random Fields," *International Conference on Pattern Recognition*, pp. 108–112, 2006.
- [70] M. Soriano, S. Huovinen, B. Martinkauppi, and M. Laaksonen, "Skin Detection in Video Under Changing Illumination Conditions," *In Proceeding International Conference on Pattern Recognition*, pp. 839–842, 2000.

- [71] V. Veznevets, V. Sazonov, and A. Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques," *In Proceeding of the GraphiCon*, pp. 85–92, 2003.
- [72] S. K. Singh, D. S. Chauhan, M. Vatsa, and R. Singh, "A Robust Skin Color Based Face Detection Algorithm," *Journal of Science and Engineering, Vol. 6, No. 4*, pp. 227–234, 2003.
- [73] R. L. Hus, M. Abdel-Mottaleb, and A. K. Jain, "Face Detection in Color Images," *IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 24(5)*, pp. 696–706, 2002.
- [74] J. Appenrodt, A. Al-Hamadi, M. Elmezain, and B. Michaelis, "Data Gathering for Gesture Recognition Systems Based on Mono Color-, Stereo Color and Thermal Cameras," *International Mega-Conference on Future Generation Information Technology, Lecture Notes in Computer Science (LNCS 5899)*, Springer-Verlag Berlin Heidelberg, pp. 78–86, 2009.
- [75] J. Appenrodt, A. Al-Hamadi, and B. Michaelis, "Data Gathering for Gesture Recognition Systems Based on Mono Color-, Stereo Color and Thermal Cameras," *International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 3, No. 1,* pp. 37–49, 2010.
- [76] P. G. R. Bumblebee, "<http://www.ptgrey.com/products/bumblebee/index.html>."
- [77] D. Scharstein and R. Szeliski, "<http://vision.middlebury.edu/stereo/>."
- [78] A. Fusiello, V. Rberto, and E. Trucco, "Efficient stereo with multiple windowing," *In CVPR*, pp. 858–863, 1997.
- [79] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 9*, pp. 920–932, 1994.
- [80] J. Mulligan and K. Daniilidis, "Predicting disparity windows for real-time stereo," *Lecture notes in computer science, 1842*, pp. 220–235, 2000.
- [81] T. Kohonen, "The Self-Organizing Map," *Proceedings of the IEEE, Vol. 78, No. 9*, pp. 1464–1480, 1990.
- [82] C. Williams and D. Barber, "Bayesian Classification With Gaussian Processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 12*, pp. 1342–1351, 1998.
- [83] M. J. Jones and J. M. Rehg, "Statistical Color Models with Application to Skin Detection," *International Journal of Computer Vision, Vol. 46, No. 1*, pp. 81–96, 2002.

-
- [84] B. Menser and M. Wien, "Segmentation and Tracking of Facial Regions in Color Image Sequences," *Proceeding of SPIE, Vol. 4067*, pp. 731–740, 2000.
- [85] C. Bishop, "Neural Networks for Pattern Recognition," *Oxford University Press*, 1995.
- [86] Y. Raja, S. J. Mckenna, and S. Gong, "Colour Model Selection and Adaptation in Dynamic Scenes," *In Proceedings European Conference on Computer Vision*, pp. 460–474, 1998.
- [87] R. A. Redner and H. F. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Review, Vol. 26, No. 2*, pp. 195–239, 1984.
- [88] M. H. Yang and N. Ahuja, "Gaussian Mixture Model of Human Skin Color and Its Applications in Image and Video Databases," *In SPIE/EI&T Storage and Retrieval for Image and Video Databases*, pp. 458–466, 1999.
- [89] X. D. Huang, Y. Ariki, and M. Jack, "Hidden Markov Models for Speech Recognition," *Edinburgh University Press*, 1990.
- [90] M. Elmezain, A. Al-Hamadi, G. Krell, S. El-Etriby, and B. Michaelis, "Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models," *IEEE International Symposium on Signal Processing and Information Technology*, pp. 1192–1197, 2007.
- [91] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, "A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition," *International Journal of Electrical, Computer, and Systems Engineering, Vol. 3, No. 3, ISSN: 2070-3813*, pp. 156–163, 2009.
- [92] S. Goronzy, "Robust Adaptation to Non-Native Accents in Automatic Speech Recognition," *Lecture Notes in Computer Sciences, Springer, ISBN-13: 978-540003250*, 2002.
- [93] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences," *Journal of WSCG, Vol.16, No. 1, ISSN: 1213-6972*, pp. 65–72, 2008.
- [94] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, "A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition," *International Conference on Computer Vision, Image and Signal Processing, PWASET, Vol. 31, ISSN: 2070-3740*, pp. 394–401, 2008.

- [95] ———, “A Hidden Markov Model-Based Continuous Gesture Recognition System for Hand Motion Trajectory,” *International Conference on Pattern Recognition*, pp. 519–522, 2008.
- [96] A. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm,” *IEEE Transactions on Information Theory*, Vol. 13, No. 2, pp. 260–269, 1967.
- [97] G. D. Forney, “The Viterbi Algorithm,” *Proceedings of the IEEE*, Vol. 61, pp. 168–278, 1973.
- [98] 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, pp. 164–171, 1970.
- [99] H. M. Wallach, “Conditional Random Fields: An Introduction,” *Technical Report MS-CIS-04-21, Univ. of Pennsylvania*, 2004.
- [100] A. McCallum, “Efficiently Inducing Features of Conditional Random Fields,” *Conference on Uncertainty in AI*, 2003.
- [101] C. Sminchisescu, A. Kananujia, and D. Metaxas, “Conditional Models for Contextual Human Motion Recognition,” *Journal of CVIU*, Vol. 104, No. 2, pp. 210–220, 2006.
- [102] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, “Hidden Conditional Random Fields for Phone Classification,” *Proceeding of European Conference on Speech Communication and technology*, pp. 1117–1120, 2005.
- [103] L. P. Morency, A. Quattoni, and T. Darrell, “Latent-Dynamic Discriminative Models for Continuous Gesture Recognition,” *IEEE Conference on CVPR*, pp. 1–8, 2007.
- [104] S. Kullback, “Information Theory and Statistics,” *Dover Publications, Inc., New York*, 1968.
- [105] T. M. Cover and J. A. Thomas, “Entropy, Relative Entropy and Mutual Information,” *Elements of Information Theory*, pp. 12–49, 1991.
- [106] M. Hwang, X. Huang, and F. Alleva, “Predicting Unseen Triphones with Senones,” *IEEE Transactions on Speech and Audio Processing*, Vol. 4, No. 6, pp. 412–419, 1996.
- [107] G. Fung, “A Comprehensive Overview of Basic Clustering Algorithms,” *IEEE Proceeding*, pp. 1–37, 2001.

-
- [108] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, "Advances in Knowledge Discovery and Data Mining," *AAAI/MIT Press*, 1996.
- [109] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression," *Kluwer Academic, Boston*, 1992.
- [110] R. O. Duda and P. Hart, "Pattern Classification and Scene Analysis," *John Wiley & Sons, New York*, 1973.
- [111] C. Ding and X. He, "k-means Clustering via Principal Component Analysis," *International Conference on Machine Learning*, pp. 225–232, 2004.
- [112] U. M. Fayyad, C. Reina, and P. S. Bradley, "Initialization of Iterative Refinement Clustering Algorithms," *In Knowledge Discovery and Data Mining*, pp. 194–198, 1998.
- [113] S. L. Phung, A. Bouzerdoum, , and D. Chai, "A Novel Skin Color Model in YC_bC_r Color Space and its Application to Human Face Detection," *IEEE International Conference on Image Processing*, pp. 289–292, 2002.
- [114] S. Askar, Y. Kondratyuk, K. Elazouzi, P. Kauff, , and O. Schreer, "Vision-Based Skin-Colour Segmentation of Moving Hands for Real-Time Application," *1st European on CVMP*, pp. 79–85, 2004.
- [115] M. Elmezain, A. Al-Hamadi, O. Rashid, and B. Michaelis, "Posture and Gesture Recognition for Human-Computer Interaction," *In-Tech Olajnica 19/2, 32000 Vukovar, Croatia, "Advanced Technologies", ISBN: 978-953-307-009-4, Book Chapter, Edited by Kankesu Jayanthakumaran*, pp. 415–440, 2009.
- [116] R. Niese, A. Al-Hamadi, and B. Michaelis, "A Novel Method for 3D Face Detection and Normalization," *Journal of Multimedia, ISSN: 1796-2048, Vol. 2, No. 5*, pp. 1–12, 2007.
- [117] R. Gonzalez and E. Woods, "Digital Image Processing," (2nd Edition). *s.l. : Prentice Hall*, 2002.
- [118] D. H. Kim and M. J. Kim, "A Curvature Estimation for Pen Input Segmentation in Sketch-based Modeling," *Computer-Aided Design, Vol. 38, No. 3*, pp. 238–248, 2006.
- [119] A. Al-Hamadi, O. Rashid, and B. Michaelis, "Posture Recognition using Combined Statistical and Geometrical Feature Vectors based on SVM," *International Journal of Computational Intelligence, Vol. 6, No. 1, ISSN: 2070-3821*, pp. 7–14, 2010.

- [120] L. Jin, C. Chen, L. Zhen, and J. Huang, "Real-Time Fingertip Detection from Cluttered Background for Vision-based HCI," *Journal of Communication and Computer*, Vol. 2, pp. 1–8, 2005.
- [121] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, pp. 564–577, 2003.
- [122] M. Elmezain, A. Al-Hamadi, R. Niese, and B. Michaelis, "A Robust Method for Hand Tracking Using Mean-shift Algorithm and Kalman Filter in Stereo Color Image Sequences," *International Conference on Computer Vision, Image and Signal Processing, PWASET*, Vol. 59, pp. 355–359, 2009.
- [123] —, "A Robust Method for Hand Tracking Using Mean-shift Algorithm and Kalman Filter in Stereo Color Image Sequences," *International Journal of Information Technology*, Vol. 6, No. 1, ISSN: 2070-3961, pp. 24–28, 2010.
- [124] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "An Efficient k-means Clustering Algorithm: Analysis and Implementation," *IEEE Transaction on PAMI*, Vol. 24, pp. 881–892, 2002.
- [125] M. Elmezain, A. Al-Hamadi, S. Pathan, and B. Michaelis, "Spatio-Temporal Feature Extraction-Based Hand Gesture Recognition for Isolated American Sign Language and Arabic Numbers," *IEEE International Symposium on Image and Signal Processing and Analysis*, pp. 254–259, 2009.
- [126] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Hand Gesture Recognition Based on Combined Features Extraction," *International Conference on Machine Vision, Image Processing, and Pattern Analysis, PWASET*, Vol. 60, pp. 459–464, 2009.
- [127] —, "Improving Hand Gesture Recognition Using 3D Combined features," *International Conference on Machine Vision*, pp. 128–132, 2009.
- [128] A. Al-Hamadi, M. Elmezain, and B. Michaelis, "Hand Gesture Recognition Based on Combined Features Extraction," *International Journal of Information Technology*, Vol. 6, No. 1, ISSN: 2070-3961, pp. 1–6, 2010.
- [129] I. V. Tetko, D. J. Livingstone, and A. I. Luik, "Neural Network Studies. 1. Comparison of Overfitting and Overtraining," *J. Chem. Inf. Comput. Sci.*, 35, pp. 826–833, 1995.
- [130] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Discriminative Models-Based Hand Gesture Recognition," *International Conference on Machine Vision*, pp. 123–127, 2009.

- [131] L.-P. Morency, A. Quattoni, C. M. Christoudias, and S. Wang, “Hidden-state Conditional Random Field Library: Version 1.3c, http://pt.sourceforge.jp/projects/sfnet_hcrf/,” 2008.
- [132] E. T. Jaynes, “The Maximum Entropy Formalism Where Do We Stand on Maximum Entropy?” (*R.D. Levine and Myron Tribus, Eds.*), *The MIT Press, Cambridge, Massachusetts*, pp. 15–118, 1979.
- [133] M. Elmezain, A. Al-Hamadi, and B. Michaelis, “A Novel System for Automatic Hand Gesture Spotting and Recognition in Stereo Color Image Sequences,” *Journal of WSCG, Vol.17, No. 1, ISSN: 1213-6972*, pp. 89–96, 2009.
- [134] —, “Hand Trajectory-Based Gesture Spotting and Recognition Using HMM,” *IEEE International Conference on Image Processing*, pp. 3577–3580, 2009.
- [135] —, “Hand Gesture Spotting Based on 3D Dynamic Features Using Hidden Markov Models,” *International Symposium on Signal Processing, Image Processing and Pattern Recognition, CCIS 61, Springer-Verlag Berlin Heidelberg*, pp. 9–16, 2009.
- [136] —, “A Robust Method for Hand Gesture Segmentation and Recognition Using Forward Spotting Scheme in Conditional Random Fields,” *International Conference on Pattern Recognition (ICPR)*, pp. 3850–3853, 2010.
- [137] R. Dugad, K. Ratakonda, and N. Ahuja, “Robust Video Shot Change Detection,” *Workshop on Multimedia Signal Processing*, pp. 376–381, 1998.

Curriculum Vitae

Name:	Mahmoud O. S. M. Elmezain
Date of Birth:	December 08, 1973; in Menofiya, Egypt
Nationality:	Egyptian
Status:	Married, three Children
E-mail:	Mahmoud.Elmezain@ovgu.de

Education:

1988 - 1992	Higher Secondary Certificate, Menofiya, Egypt.
1992 - 1996	B.Sc. in Pure Mathematics and Computer Science. Faculty of Science, Menofiya University, Egypt.
1998 - 1999	Postgraduate courses for M.Sc. Faculty of Computers and Information, Helwan University, Egypt.
2000 - 2004	M.Sc. in Computer Science. Faculty of Computers and Information, Helwan University, Egypt.
2006 - Now	Works towards Ph.D. degree at IESK, Otto-von-Guericke-University Magdeburg, Germany.

Work Experience:

1997 - 2004	Demonstrator in Dept. of Statistics and Computer Science, Faculty of Science, Tanta University, Egypt.
2004 - 2006	Assistant lecturer in Dept. of Statistics and Computer Science, Faculty of Science, Tanta University, Egypt.

Magdeburg, November 26, 2010

Mahmoud Elmezain

Related Publications

The presented thesis is based on the following international reviewed journal and conferences papers:

1. M. Elmezain, A. Al-Hamadi, B. Michaelis: **“Robust Methods for Hand Gesture Spotting and Recognition Using Hidden Markov Models and Conditional Random Fields”**, IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), December 15-18, 2010. Luxor, Egypt (Accepted)
2. S. Sadek, A. Al-Hamadi, M. Elmezain, B. Michaelis, U. Sayed: **“Human Activity Recognition via Temporal Moment Invariants”**, IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), December 15-18, 2010. Luxor, Egypt (Accepted)
3. M. Elmezain, A. Al-Hamadi, B. Michaelis: **“A Robust Method for Hand Gesture Segmentation and Recognition Using Forward Spotting Scheme in Conditional Random Fields”**, International Conference on Pattern Recognition (ICPR), pp. 3850-3853, Aug. 23-26, 2010. Istanbul, Turkey
4. A. Al-Hamadi, M. Elmezain, B. Michaelis: **“Hand Gesture Spotting Based on 3D Dynamic Features Using Hidden Markov Models”**, International Journal of Computational Intelligence, Vol. 6, No. 1, ISSN: 2070-3821, pp. 1-6, 2010. Paris, France
5. M. Elmezain, A. Al-Hamadi, R. Niese, B. Michaelis: **“A Robust Method for Hand Tracking Using Mean-shift Algorithm and Kalman Filter in Stereo Color Image Sequences”**, International Journal of Information Technology, Vol. 6, No. 1, ISSN: 2070-3961, pp. 24-28, 2010. Paris, France
6. M. Elmezain, A. Al-Hamadi, B. Michaelis: **“Improving Hand Gesture Recognition Using 3D Combined features”**, International Conference on Machine Vision (ICMV), pp. 128-132, December 28-30, 2009. Dubai, UAE
7. M. Elmezain, A. Al-Hamadi, B. Michaelis: **“Discriminative Models-Based Hand Gesture Recognition”**, International Conference on Machine Vision (ICMV), pp. 128-132, December 28-30, 2009. Dubai, UAE

8. M. Elmezain, A. Al-Hamadi, B. Michaelis: **“Hand Gesture Spotting Based on 3D Dynamic Features Using Hidden Markov Models”**, International Conference on Machine Vision, Image Processing, and Pattern Analysis (MVIIPA), PWASET, Vol. 60, pp. 459-464, December 25-27, 2009. Bangkok, Thailand
9. M. Elmezain, A. Al-Hamadi, B. Michaelis: **“Hand Gesture Spotting Based on 3D Dynamic Features Using Hidden Markov Models”**, International Symposium on Signal Processing, Image Processing and Pattern Recognition (SIP), CCIS 61, Springer-Verlag Berlin Heidelberg, pp. 9-16, December 10-12, 2009. Jeju Island, Korea
10. J. Appenrodt, A. Al-Hamadi, M. Elmezain, B. Michaelis: **“Data Gathering for Gesture Recognition Systems Based on Mono Color-, Stereo Color and Thermal Cameras”**, International Mega-Conference on Future Generation Information Technology (FGIT), Lecture Notes in Computer Science (LNCS 5899), Springer-Verlag Berlin Heidelberg, pp. 78-86, December 10-12, 2009. Jeju Island, Korea
11. M. Elmezain, A. Al-Hamadi, R. Niese, B. Michaelis: **“A Robust Method for Hand Tracking Using Mean-shift Algorithm and Kalman Filter in Stereo Color Image Sequences”**, International Conference on Computer Vision, Image and Signal Processing (CVISP), PWASET, Vol. 59, pp. 355-359, November 25-27, 2009. Bali, Indonesia
12. M. Elmezain, A. Al-Hamadi, B. Michaelis: **“Hand Trajectory-Based Gesture Spotting and Recognition Using HMM”**, IEEE International Conference on Image Processing (ICIP), pp. 3577-3580, November 7-11, 2009. Egypt
13. M. Elmezain, A. Al-Hamadi, J. Appenrodt, B. Michaelis: **“A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition”**, International Journal of Electrical, Computer, and Systems Engineering, Vol. 3, No. 3, ISSN: 2070-3813, pp. 156-163, 2009. Paris, France
14. M. Elmezain, A. Al-Hamadi, S. Pathan, B. Michaelis: **“Spatio-Temporal Feature Extraction-Based Hand Gesture Recognition for Isolated American Sign Language and Arabic Numbers”**, IEEE International Symposium on Image and Signal Processing and Analysis (ISPA), pp. 254-259, September 6-18, 2009. Salzburg, Austria
15. M. Elmezain, A. Al-Hamadi, Omer Rashid, B. Michaelis: **“Posture and Gesture Recognition for Human-Computer Interaction”**, In-Tech Olajnica

19/2, 32000 Vukovar, Croatia, "Advanced Technologies", ISBN: 978-953-307-009-4, Book Chapter, Edited by Kankesu Jayanthakumaran, pp. 415-440, October 2009.

16. M. Elmezain, A. Al-Hamadi, B. Michaelis: "**A Novel System for Automatic Hand Gesture Spotting and Recognition in Stereo Color Image Sequences**", Journal of WSCG, Vol.17, No. 1, ISSN: 1213-6972, pp. 89-96, Feb. 2-5, 2009. Plzen, CZ
17. S. Pathan, A. Al-Hamadi, M. Elmezain, B. Michaelis: "**Feature-supported Multi-hypothesis Framework for Multi-object Tracking using Kalman Filter**", International Conference on Computer Graphics, Visualization and Computer vision , WSCG, pp. 197-202, Feb. 2-5, 2009. Plzen, CZ.
18. M. Elmezain, A. Al-Hamadi, B. Michaelis: "**Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences**", Journal of WSCG, Vol.16, No. 1, ISSN: 1213-6972, pp. 65-72, February 4-7, 2008. Plzen, CZ
19. M. Elmezain, A. Al-Hamadi, J. Appenrodt, B. Michaelis: "**A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition**", International Conference on Computer Vision, Image and Signal Processing (CVISP2008), PWASET, Vol. 31, ISSN: 2070-3740, pp. 394-401, Juli 25-27, 2008. Prague, CZ
20. M. Elmezain, A. Al-Hamadi, J. Appenrodt, B. Michaelis: "**A Hidden Markov Model-Based Continuous Gesture Recognition System for Hand Motion Trajectory**", International Conference on Pattern Recognition (ICPR 2008), pp. 519-522, Dec. 8-11, 2008. Tampa, Florida, USA
21. M. Elmezain, A. Al-Hamadi, G. Krell, S. El-Etriby, B. Michaelis: "**Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models**", IEEE International Symposium on Signal Processing and Information Technology, pp. 1192-1197, December 15-18, 2007. Cairo, Egypt