ALIGNMENT AND MULTIMODAL ANALYSIS IN SIGNED SPEECH

by

Pınar Santemiz

B.S, in Mathematics, Boğaziçi University, 2006

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2009

ALIGNMENT AND MULTIMODAL ANALYSIS IN SIGNED SPEECH

APPROVED BY:

Prof. Lale Akarun .....
(Thesis Supervisor)

Assist. Prof. Murat Saraçlar .....

Assoc. Prof. Pınar Yolum .....

DATE OF APPROVAL: 20.07.2009

ACKNOWLEDGEMENTS

I wish to express my gratitude to all the people who have given me encouragement and helped me in the completion of this study. Especially I would like to thank my supervisor Prof. Lale Akarun for her invaluable guidance, academic feedback, support and patience during this thesis. She was the one who impressed and inspired me in such a way that I had decided to continue my studies in computer engineering department, while I was an undergraduate student in mathematics. I would like to thank Assist. Prof. Murat Saraçlar for his valuable ideas and comments, and to Assoc. Prof. Pınar Yolum for participating in my thesis jury and giving me valuable feedback. I also wish to thank Prof. Bülent Sankur and Prof. Ethem Alpaydın for fruitful discussions and academic help, and to all my teachers who influenced me throughout my education.

I am especially indebted to my parents, Güner Santemiz and Nilüfer Santemiz, for all their love, support, and encouragement. I also would like to thank my brother Levent Santemiz for being an inspiration, and her wife Zeynep Konan for her invaluable aid with my presentation. My lovely companion, Tappity, deserves also a special thank for correcting my spelling with her paws and always reminding me the time.

I would like to present my gratitude to my mentor Oya Aran not only for her support, encouragement and contributions, but also for being a role model with her great personality, her reliable judgement and her coveted rhythm. My special thanks go to Berk Gökberk with whom I feel very lucky to have met in my life. Without him, I would not be able to continue so easily when things were not going well, and also I would not enjoy this work as much.

I would like to thank the members of Satellite Laboratory, Birkan Yılmaz, Şükrü Kuran, Evren Önem, Fernaz Alimoğlu, Suzan Bayhan, Bora Zeytinci, Gürkan Gür, Onur Türkyılmaz, and Kaan Bür for providing such an amusing environment. I would like to thank the members of Perceptual Intelligence Laboratory, my spiritual twin, Neşe Alyüz for accompanying me in every possible way, İsmail Arı for his invaluable

help and humble personality, Cem Keskin for sharing his wisdom and our staring contests, Yunus Emre Kara and Gaye Genç for their sincere friendship. I also thank all other friends for their support and for sharing their experiences: Albert Ali Salah, Onur Dikmen, Koray Balcı, Atay Özgövde, Rabun Koşar, Erinç Dikici, Hamdi Dibeklioğlu, Dağhan Dinç, and Umut Konur.

This study has been supported by TÜBİTAK under project 107E021.

ABSTRACT

ALIGNMENT AND MULTIMODAL ANALYSIS IN SIGNED SPEECH

In this thesis, we attack the problem of extracting isolated signs from continuous signed speech videos. Signed speech is a language that uses the signs of sign language and the grammar of spoken language. It is a visual language and makes use of hand gestures, which consist of hand motion and hand shape. In continuous signed speech, signs are expressed in succession, which results in coarticulation effects, making segmentation a challenging task. In this work, we aim to segment some of the most common signs in Turkish Sign Language using hand gesture information. This process consists of two consecutive steps: First, we apply segmentation to hand regions obtained from a hand tracking module and obtain images containing only the left or the right hand. Then, we represent hand gestures by a variety of features which can be categorized as follows: 1) Center of mass coordinates of each hand and their first-order derivatives, 2) Ellipse parameters for each hand, 3) Discrete Cosine Transform, 4) Histogram of oriented gradients, 5) Local Binary Patterns, 6) Hu Moments, and 7) Radial Distances. Then, we align the sequences with different methods and find the start and end positions for each sign. We use Dynamic Time Warping (DTW), Hidden Markov Models (HMM), and coupled HMMs as different alignment approaches. We also apply some fusion techniques to improve the alignment performances. We experiment on a database from Turkish signed speech videos and report the results. We see that the highest accuracy is obtained by combining the DTW and HMM methods using Center of mass coordinates, their first-order derivatives, and Ellipse features.

ÖZET

İŞARET DİLİ VİDEOLARINDAN AYRIK İŞARET ÇIKARIMI

Bu tezde, duraklamasız, işaretlenmiş Türkçe videolarından otomatik ayrık işaret çıkarımı yapan bir yöntem geliştirdik. İşaretlenmiş Türkçe, Türk işaret dilindeki işaretlerin, Türkçe konuşma diline ait dilbilgisi kuralları dahilinde kullanılmasından oluşmuş görsel bir dildir ve el şekli ile el hareketlerinden oluşur. Duraklamasız işaret dilinde peş peşe yapılan işaretler birbirlerini etkileyebilir ve işaretlerin başında ve sonunda değişimler olabilir. Bu da ayrık işaret çıkarımını zorlaştırır. Bu çalışmada, el işaretleri bilgisini kullanarak Türk işaret dilinde sıklıkla kullanılan bazı işaretlerin ayrık çıkarımını yapmayı amaçladık. Kullandığımız yöntem iki adımdan oluşmaktadır: İlk olarak, el takibi modülü kullanılarak elde edilmiş el bölgelerine bölütleme uygulayıp sadece sağ veya sol eli içeren resimler elde ettik. Daha sonra el işaretlerini çeşitli yöntemler kullanarak betimledik. Bunlar şu şekilde sıralanabilir: 1) Ellerin merkez koordinatları ve bu koordinatların türevleri, 2) Her bir el için elips değişkenleri, 3) Ayrık kosinüs dönüşümü, 4) Yönlü Gradyan Histogramı, 5) Yerel İkili Örüntü yöntemi, 6) Hu momentleri, ve 7) Işınsal Uzaklık fonksiyonu. Daha sonra işaret dizilerini çeşitli yöntemler kullanarak hizalayıp işaretlerin başlangıç ve bitiş noktalarını bulduk. Hizalama yöntemleri olarak dinamik zaman bükmesi (DTW), saklı Markov modeli (HMM) ve bağlaştırılmış HMM kullandık. Ayrıca sistemimizin başarımını arttırmak amacıyla bazı tümleştirme yöntemleri uyguladık. Sistemimizin başarımını işaretlenmiş Türkçe videolarından oluşan bir veri tabanı üzerinde gösterdik. Buna göre, DTW ile HMM algoritmaları birleştirildiğinde ve öznitelik olarak merkez koordinatları, bu koordinatların türevleri, ve elips değişkenleri kullanıldığında en yüksek başarımı elde ettik.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT v
ÖZET vi
LIST OF FIGURES ix
LIST OF TABLES xi
LIST OF SYMBOLS/ABBREVIATIONS xiii
1. INTRODUCTION 1
1.1. Motivation 1
1.2. Literature Review 2
1.2.1. Hand Tracking 2
1.2.2. Feature Extraction 3
1.2.3. Classification 7
1.3. Approach and Contributions 13
1.4. Outline of the Thesis 15
2. FEATURE EXTRACTION 18
2.1. Segmentation of the Hand 18
2.1.1. Region Growing 20
2.1.2. Template Matching 21
2.1.3. Elimination of Poor Segmentation 22
2.2. Center of Mass (CoM) 23
2.3. Ellipse (E) 24
2.4. Radial Distance Function (RDF) 24
2.5. HU Moments 25
2.6. Discrete Cosine Transform (DCT) 27
2.7. Histogram of Oriented Gradients (HOG) 29
2.8. Local Binary Patterns (LBP) 31
2.9. Postprocessing of the Features 32
3. ALIGNMENT TECHNIQUES 37
3.1. Dynamic Time Warping (DTW) 37

3.2. Hidden Markov Models (HMM) 42
3.2.1. Left-Right Hidden Markov Models 43
3.2.2. Coupled Hidden Markov Models 48
3.3. Fusion Techniques 51
4. EXPERIMENTAL RESULTS 53
4.1. Database 53
4.2. Alignment 55
4.3. Recognition 67
5. CONCLUSION 77
REFERENCES 80

LIST OF FIGURES

Figure 1.1. An example frame from the news recordings. 15

Figure 1.2. General system flow. 16

Figure 2.1. Feature extraction procedure 19

Figure 2.2. Phases of preprocessing 20

Figure 2.3. Output of segmentation 23

Figure 2.4. A representation of the CoM and E parameters 24

Figure 2.5. RDF configuration procedure 25

Figure 2.6. The procedure of zig-zag scanning 28

Figure 2.7. The procedure to obtain DCT features 29

Figure 2.8. Sample outputs of the HOG method 31

Figure 2.9. An example of computing LBP 32

Figure 2.10. Uniform patterns encoded in LBP 33

Figure 2.11. A sample output of the LBP method 33

Figure 2.12. Moving average filter algorithm 34

Figure 3.1. Construction of accumulated distance matrix 39

Figure 3.2. Alignment with DTW 40

Figure 3.3. Segmentation with DTW 42

Figure 3.4. Left-to-right HMM 43

Figure 3.5. Viterbi Algorithm 47

Figure 3.6. Dependency graph of a bimodal coupled HMM 48

Figure 3.7. Dependency graph of HMM 50

Figure 4.1. Occlusion examples in the database 55

Figure 4.2. The True Positive (TP), True Negative (TN), False Postive (FP) and False Negative (FN) values 57

Figure 4.3. Precision-recall curve 59

Figure 4.4. Six examples for the alignment of the word “prime minister” for DTW and HMM 64

LIST OF TABLES

Table 1.1. Some gesture recognition techniques 14

Table 3.1. Feature sets obtained by feature level fusion 51

Table 3.2. Feature sets used in cHMM and parallel HMM 52

Table 4.1. Database 56

Table 4.2. Effect of enlargement window 58

Table 4.3. Performance of using only the right hand and the two hands 61

Table 4.4. Performance of DTW and HMM in the enlarged window $(+15, -1)$ 62

Table 4.5. Performance of DTW and HMM in the enlarged window $(+21, +6)$ 68

Table 4.6. Performance of coupled and parallel HMMs in the enlarged window $(+15, -1)$ 69

Table 4.7. Performance of coupled and parallel HMMs in the enlarged window $(+21, +6)$ 70

Table 4.8. Performance of DTW-HMM in the enlarged windows $(+15, -1)$ and $(+21, +6)$ 71

Table 4.9. Performance of coupled and parallel HMMs after DTW with the enlarged window $(+15, -1)$ 72

Table 4.10. Performance of coupled and parallel HMMs after DTW with the enlarged window $(+21, +6)$ 73

Table 4.11. The effect of occlusion, and number of hands involved in signing . 74

Table 4.12. The effect of duration 75

Table 4.13. Recognition performances 76

LIST OF SYMBOLS/ABBREVIATIONS

a_{ij}	Transition probability from state i to state j
$\mathbf{A}$	State transition matrix of a hidden Markov model
b_j	Observation probability of state i
$\mathbf{B}$	Observation probabilities of a hidden Markov model
$\mathbf{D_A}$	Accumulated distance matrix
$\mathbf{D_L}$	Local score matrix in DTW
f	Frame number
F	Total number of frames in a video sequence
g_H	Horizontal gradient value
g_V	Vertical gradient value
$I(x, y)$	Intensity value of the pixel at the coordinates x and y
L	Length of Gaussian window
l_m	m^th Local maximum location
m_{pq}	Two-dimensional $(p + q)^{th}$ order image moment
m^C_{pq}	Central image moment of order $(p + q)$
M	Height of the image
N	Width of the image
O_t	Observation at time t
P_W	Alignment Path
q^*	optimal path
q_t	state variable
Q	Number of states in a hidden Markov model
s_i	Probability of being in state i in HMM
S	Dynamic time warping score
$\mathbf{v^f}$	Feature vector of frame f
$\mathbf{v_{DCT}}$	DCT feature vector
$\mathbf{v_{HOG}}$	HOG feature vector
$\mathbf{v_{Hu}}$	Hu moment feature vector
v_m	Distinct observation symbol

$\mathbb{V}$	Alphabet of distinct observation symbols
x	x coordinate value
y	y coordinate value
α	Forward variable
β	Backward variable
η_{pq}	Normalized central image moment of order $(p+q)$
$\mathfrak{L}$	Likelihood
λ	Eigenvalue
Λ	The proportion of variance
μ_i	Mean vector of state i
ω	Gaussian window
π_i	Initial probability of state i
Π	Initial state probabilities of a hidden Markov model
ρ	Density distribution function
Σ_i	The covariance matrix of state i
$\sigma(.)$	The standard deviation of the ith DCT coefficient
Θ	Gradient orientation
Acc	Accuracy
cHMM	Coupled Hidden Markov Models
CoM	Center of Mass
DCT	Discrete Cosine Transform
DTW	Dynamic Time Warping
DTW-HMM	Sequential Fusion of DTW and HMM
DTW-cHMM	Sequential Fusion of DTW and coupled HMM
DTW-pHMM	Sequential Fusion of DTW and parallel HMM
E	Ellipse Parameters
EM	Expectation Maximization
FN	False Negative
FP	False Positive

GT	Ground Truth
H	Hu moments
HCI	Human-Computer Interaction
HMM	Hidden Markov Models
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
Ovr	Overlapping Ratio
pHMM	Parallel Hidden Markov Models
Pre	Precision
Rec	Recall
RGB	Red Green Blue
SLR	Sign Language Recognition
TN	True Negative
TP	True Positive
TSL	Turkish Sign Language
2D	Two Dimensional
3D	Three Dimensional

1. INTRODUCTION

Sign languages are the natural communication media of the hearing-impaired. They are visual languages which make use of hand gestures, body movements and facial expressions. These modalities are expressed together to form a sign. Each sign corresponds to a word in spoken languages.

Sign languages are well-structured languages having their own morphology, syntax and grammar. Since they make use of different modalities, the linguistic characteristics of sign language are different than those of spoken languages. In addition to sign language, there is another type of visual communication called "signed speech". Here, for each word, the corresponding sign in sign language is used but the word ordering is used as in spoken language.

As a result of the complexity of the visual analysis of hand gestures and the multimodal nature of sign language, little advance has been observed despite the growing interest in the field of sign language analysis and understanding in the past 10 years.

1.1. Motivation

Sign Language Recognition (SLR) can be classified as two different problems according to the available data in the input: isolated or continuous recognition. In isolated recognition, the aim is to recognize a single gesture. Therefore, the start and end conditions of each of the gestures are predefined and the problem can be considered as differentiating the gesture that needs to be recognized.

When signs are performed in a continuous sequence to form sentences, the appearance of a sign is affected by the preceding and succeeding signs. This is called the co-articulation effect and makes segmentation a challenging task. Therefore, an automatic system that extracts isolated signs may be very useful for recognition tasks.

Another application of such a system is to create a sign-to-text translator. People who are interested in learning sign language need a large-vocabulary dictionary. Also SLR systems need a database of signs or phonemes for training a model that will be used in the recognition task. Hence, a system that enables this task automatically is required both by the linguistics and the computer vision community.

In this work, we are interested in aligning and extracting isolated signs in continuous sign language videos.

1.2. Literature Review

There has been a growing interest in sign language recognition and analysis for the last 10 years. A complete SLR system should handle both manual signals like hand shape, position and movement, and non-manual signals like facial expressions, head motion and body gesture [1]. Hence, it is a challenging task to reach a large-vocabulary recognition system.

Most of the sign language analysis systems include three main components. First, detection, segmentation and tracking of the hand or the body parts must be handled. Secondly, features describing the manual or non-manual information must be described. Finally, segmentation or classification of the signs is performed.

1.2.1. Hand Tracking

In tracking and segmentation of the hands, the main problem is the accurate detection and segmentation in presence of occlusion. Many hand gesture recognition and SLR systems handle this problem with the use of instrumented gloves [2–6]. Another approach is to use markers such as colored gloves [7] or stereo camera to obtain 3D locations of the hand [8,9]. Recently, real-time vision based systems have been adopted which require only one camera and use only the skin color information [10–18]. Most of these methods assume controlled environment or restricted clothing.

Non-manual information also has an important role in sign language. In the presence of facial expressions and head motion, the meaning of a sign may be strengthened, weakened or even completely changed. There are some approaches where both manual and non-manual information are exploited [17, 19, 20].

1.2.2. Feature Extraction

Manual signals are the basic components in sign language and consist of hand shape, hand motion and hand position. In order to describe these components, shape and motion features must be extracted.

Hand shape is one of the main modalities of sign language. Because of the high degree of freedom of the hand, describing hand gestures is also a challenging task for many Human-Computer Interaction (HCI) applications that use hand shapes in gesture controlled computer systems. Especially when the number of hand shapes that are used in the system is large as in sign language, and 2D images captured by a camera are used, it becomes very complex to distinguish a specific hand shape.

An efficient shape feature must have certain properties like identifiability, noise resistance, translation, rotation and scale invariance and robustness to occlusion. Moreover, the descriptors should be represented and stored compactly in order to have acceptable computation time when the similarity or the distance between the descriptors is computed. Consequently, many hand shape description techniques have been proposed. These can be grouped as model-based methods and appearance based methods.

In model-based methods, 3D models of the hand are obtained using stereo cameras and colored markers or sensor equipped devices. Then, the test data is matched to the models of known objects. To obtain the shape information of each hand, 18-sensor gloves are used [5,6]. Here, the gloves transform the hand and finger configuration into joint-angle data, and this information is used directly to recognize signs.

However, this is an expensive approach when the test data includes only a 2D

image of the hand, because in this case a large database is needed to obtain enough models for describing all possible hand shapes. Agris *et al.* used colored gloves with six differently colored visual markers, and collected a database containing a large number of postures seen from many different view angles [21]. From this database, descriptive and stable 2D features and 3D hand posture parameters are extracted in an off-line preparation phase. Then, for each frame of the test video, N postures are retrieved and at the end of the sequence, a smooth posture sequence is obtained.

In appearance based methods, instead of using 3D models objects are modeled based on how they can appear in 2D images. In most of the studies, appearance based methods are preferred due to their low cost, user-friendliness, simplicity and low computation times, especially for real time applications. These methods can be grouped according to the descriptors they use, namely, region based descriptors and texture based descriptors.

In region based gesture recognition methods, the binary image of the segmented hand is used to extract the outer contour of the hand; or simple geometric features such as the center of gravity, width and height of the principal axes, axis of least inertia, area, or image moments, are used.

Outer contour of the hand image is mainly preferred when hand segmentation accuracy is high, since contours are greatly affected by segmentation errors and occlusion. Therefore, methods relying on the outer contour are not commonly used in sign language recognition applications. On the other hand, geometric features are resistant to noise and normalization can eliminate both translation and scale variance. Since they only provide general information on the hand shape, geometric features are used to describe images with low-resolution [12], or are used in combination with other modalities like facial expressions [7, 21] or speech information [18].

To differentiate between hand shapes having similar silhouettes, texture based methods are used to describe the finger's position and orientation. These features include Local Binary Patterns (LBP), orientation histograms, Discrete Cosine Trans-

forms (DCT), and Scale-Invariant Feature Transform (SIFT).

In Local Binary Pattern (LBP) based methods, relationships between the intensity values of neighboring pixels are used to describe the local statistical distributions of the pixel values. LBP based methods are generally preferred to recognize face [22] or facial expressions [23]. Recently, a hand posture recognition approach was presented where a LBP variant method is used for describing hand shape, and it has been shown that LBP features can also be applied to the problem of hand posture recognition [24].

Orientation histograms were previously used for gesture classification and interpolation [25]. Orientation histograms describe the shape of an object by the distribution of local intensity gradient orientations or edge directions. The main advantage of local orientations is their robustness to illumination changes. Moreover, forming a histogram from the orientations gives translational invariance. Yet, orientation histograms cannot handle occluded data and therefore a successful segmentation of the hands is required.

Nayak *et al.* proposed a method which does not rely on hand tracking and precise hand segmentation [15]. Here, the edge information both at the boundary and inside the skin colored areas are used as low-level image primitives. Then, the distribution of pairwise relationships between these low-level primitives are used for sign language recognition, gesture recognition and action recognition.

Binh *et al.* used Discrete Cosine Transform (DCT) to describe hand shape for recognition of ASL letter spelling alphabet and digits [26]. DCT is often used in signal and image processing, especially for data compression. Here, the spatial representation in the image is converted into a frequency representation where low frequency components are easily detected.

Another approach to describe shape is the use of Scale-Invariant Feature Transform (SIFT) features [27]. SIFT extracts repeatable characteristic feature points from an image and generates descriptors representing the texture around the feature points. SIFT features are invariant to image scale and rotation. They are also robust to changes

in illumination, noise, and minor changes in viewpoint, which makes them also robust to partial occlusion.

For describing hand motion, the location and the velocity information of the hands are required. When stereo cameras or instrumented gloves are used, 3D location of the hands can easily be found [2–6, 9, 28].

In vision based systems using 2D information, hand locations are described by the center of mass coordinates of the segmented hand [7,10,12,16–18]. Segmentation of the hands may be erroneous due to occlusion or illumination changes. Therefore, it may be necessary to smooth the trajectories to overcome the noise resulting from segmentation error. In many approaches, in addition to the hand trajectory, velocity of the hands is also found by computing the first order derivative of the hand trajectory [7, 10, 12, 16].

Hand position has an important meaning in sign language. For example in Turkish Sign Language (TSL), the sign "ache" means "headache" or "stomachache" when the sign is positioned around the head or stomach, respectively. Relative positions of the hands with respect to each other carry significant amount of information about the sign. Signs may be one-handed where the dominant hand makes the sign and the other hand's movement does not affect the meaning, or they may be two-handed and the synchronization characteristics of the two hands may carry important information. For example, in TSL, the signs "marriage" and "divorce" have the same hand shape but the hands are approaching and moving away, respectively.

In many methods, in addition to the hand position information, the distance between the two hands and the relative distance of the hands and other body parts are computed. In many approaches, the face of the signer is taken as a reference to resolve the relation between the hands and the face [3, 12, 17, 18]. Another approach is to concatenate the hand positions and the relative hand positions to form the position feature vector [4, 5].

1.2.3. Classification

Research on sign language analysis aims to realize several tasks such as translation of sign language, creating large-vocabulary sign dictionaries or tutoring sign language with an interactive system. In order to succeed in these tasks, three components of classification must be handled: alignment, segmentation and recognition.

Alignment is a way of finding correspondences between elements of different sequences to identify regions of similarity. Alignment is usually followed by segmentation which refers to the division of sequences into a series of analogous segments. These segments may represent signs or non-sign parts in sign language sentences, or phonemes which represent the basic elements of gestures in sign language. Recognition aims to identify these video segments. In many recognition tasks, segmentation and alignment is implicitly handled. However, there are examples where these components are handled separately [9]. There are also applications that focus on sign spotting, where classification of the signs require alignment and clustering abilities.

SLR systems are differentiated according to the data they use as input. The data may contain isolated signs, where the starting and ending conditions of the gestures are known, or continuous signs where the signs are parts of a sequence and are affected by the preceding and succeeding signs.

Initial studies on SLR have concentrated on static signs, such as the finger spelling alphabet or some selected static signs, where only hand shape information is used for recognition. However, because of the dynamic nature of sign language if temporal data is discarded, only a limited vocabulary of signs can be recognized. Consequently, in later studies, researchers started to work on temporal data to recognize dynamic signs.

In order to describe the temporal characteristics of the signs, several methods are used such as Dynamic Time Warping (DTW) [9,12], neural networks [4,6], HMMs [17, 27] or dynamic Bayesian networks [29].

Kong *et al.* aimed to recognize 27 groups of hand shapes representing the letters of the English alphabet and 6 other hand shapes which are frequently used in Signing Exact English (SEE) [6]. Here, a linear decision tree with Fisher's linear discriminant is used for classification of the hand shape, and Vector Quantization Principal Component Analysis (VQPCA) is used for classification of the trajectory of the hand. In VQPCA, the data is clustered based on transformation characteristics and local PCA of each cluster is computed. The component classifiers are later combined by using a look-up table to recognize the complete sign.

Some isolated gesture recognition systems use DTW based methods to compare temporal characteristics of the gestures [12, 30]. DTW is a widely used approach for aligning signals and computing the similarity. It is a well studied method in the field of bioinformatics where several sequence alignment methods are used to align DNA, RNA sequences, and proteins for database searches or structure prediction of protein families [31, 32]. DTW has also been applied for online or offline signature verification [33, 34] and speech recognition tasks [35].

Corradini *et al.* recognized human hand-arm movements from a small vocabulary using DTW [12]. Here, DTW is used for computing the distance between an unknown input and a set of previously defined templates. Lichtenauer *et al.* propose a method to construct a classification model for a new sign using information of known classifiers trained for other signs [30]. When a classifier for a new sign needs to be constructed, first, synchronization between the new sign and the signs in the database are performed using DTW. Then, the parameters of the classifier are derived by generalizing the comparable parts of the models of different classes.

Although there are studies that use DTW based methods for isolated recognition, HMMs generally show better performances and are easily applied to continuous recognition tasks. In the continuous case, the success of HMMs comes from the fact that they can implicitly segment continuous sequences.

Starnet *et al.* presented a real-time continuous American Sign Language (ASL)

recognition method using HMM [10]. Here, a 40-word lexicon is recognized, where the tracking process produces only a coarse description of hand shape, orientation and trajectory.

Aran *et al.* developed an interactive system where signing of users is classified and their performance is evaluated [7, 19]. This system uses multi-modal analysis of the hand motion and shape analysis together with head motion analysis using hidden Markov models. In order to simplify problems such as occlusion, colored gloves are used. In another study, an improved hand tracking method is proposed, where skin color information is used and situations including fast movement, interactions and occlusions are robustly handled [18, 19]. Further multi-modal fusion techniques are investigated to combine manual and non-manual components [19, 36]. Here, a two-step sequential belief based fusion strategy is proposed, which enables to decide on the reliability of the components.

Agris *et al.* used manual and facial features for both isolated and continuous sign language recognition [17]. Classification is based on HMMs. Manual and facial features are classified separately, and then, results are merged for recognition.

There are also studies that successfully apply HMM for the isolated gesture recognition task. Liu *et al.* compared HMMs using fully-connected, left-right and left-right banded model structures for isolated hand gesture recognition [13].

In some approaches, both HMM and DCT are used to train models. Lichtenauer *et al.* separated time warping and classification [9]. Here, statistical DTW (SDTW) is used to model a reference by iteratively warping all training samples with an initial model and reestimating the model parameters. The reference is a statistical model consisting of a normal distribution for each time point. Then, for classifying the warped features Combined Discriminative Feature Detectors (CDFD) and Quadratic Classification on DF Fisher Mapping (Q-DFFM) are used. The performance of the system is shown on a database containing 3D hand motion features of 120 different signs from Dutch Sign Language collected using a calibrated stereo camera.

Alon *et al.* proposed a system where a gesture can be recognized even when the hand location is highly ambiguous [16]. The method can be applied to continuous image streams with moving, cluttered backgrounds. Here, both DTW and HMM are used to train a model. The method is evaluated in three settings. In the first setting, gestures corresponding to the 10 digits are performed where the signer can wear short sleeved shirts and the environment is not controlled. In the second setting, sign retrieval of three ASL signs from a 15 minute long ASL story is aimed. In the third setting, video clips of ASL signs containing 24 signs are used where each sign starts and ends with a particular hand shape and the signer wears colored gloves.

Since sign language consists of multiple interacting processes, extensions on HMMs such as coupled HMMs [28] and parallel HMMs [3] have also been proposed. These methods are formulated to solve general problems of time series modeling and classification under multiple observation data sets.

Coupled HMMs are mostly used for audio-visual speech recognition. Nefian *et al.* proposed a system that couples lip information with speech to recognize 36 isolated words in the CMU database [37]. The performances of coupled HMM and factorial HMM are compared for audio-visual isolated word recognition and it is shown that cHMM outperforms factorial HMM [38].

Brand *et al.* tested performance of coupled HMMs with upper-body gesture recognition, where each arm is treated as a process [8]. The performance of the system is tested with 3D hand-tracking from a Chinese meditative exercise. Later, the performance of conventional HMMs, linked HMMs and coupled HMMs are compared on the same database and it is shown that the cHMMs outperform the other models [28].

Pavlovic *et al.* compared three complex Dynamic Bayesian Network (DBN) models are compared: mixtures of DBNs, mixed-state DBNs, and coupled HMMs [29]. The system's performance is shown on a database containing 11 isolated gestures to describe 40 gestural commands. Here, it is shown that coupling between speech and gestures improves the performance significantly, when compared to independent classification

of speech and gestures.

Later, parallel HMMs for ASL recognition is presented, where 3D data of continuous American sign language is used and the movements of the left and right hand are modeled using parallel HMMs [3]. Parallel HMMs are independent HMMs with separate outputs. After training the HMMs independently, the probabilities of the HMMs are combined at word or phoneme ends.

Although sign segmentation is implicitly handled in continuous recognition tasks, direct sign segmentation or spotting is rarely addressed. Recently, a discriminative method for sign spotting to align signed English is presented [27]. The authors first apply a rough alignment via HMMs, followed by sign spotting based on a discriminative model.

Cooper *et al.* performed sign spotting of 23 signs from continuous ASL [39]. Here, subtitles are used to find possible search intervals, where the intervals are 200 frames long. The searched sign is assumed to be of fixed length with 10 frames of duration. To find the potential sign positions, mean shift algorithm is performed. Then, using temporally constrained apriori mining algorithm, target locations are found.

Buehler *et al.* used continuous British sign language videos for segmentation of 210 signs [40]. Here, HOG features and position of the hands are used to describe the hand gesture. The features are concatenated for 7-13 frames. Then, using sliding window classifier, the target locations of the signs are found. The segmentation result is accepted, if a temporal overlap of at least 50 per cent with the ground truth is achieved.

Nayak *et al.* presented segmentation of 10 signs from continuous signed English videos [41]. Here, instead of performing hand tracking, the edge information in skin colored blobs are used. For extracting the signs, all possible substrings are aligned DTW, and iterated conditional modes are used to find the target locations. All the extracted patterns are examined by an ASL expert, and if she decides that the segmentation is

correct, the result is accepted.

Sign segmentation is also used as a part of a complete recognition method in continuous SLR. Lee *et al.* proposed a threshold model for gesture spotting to handle non-gesture patterns [11]. The threshold model is a discrete left-to-right HMM and it is constructed by combining all the gesture models in the system. Using the threshold model, trained gestures are extracted from continuous hand motion. In a later study, it is aimed to recognize whole body key gestures [42]. In order to do that, a garbage gesture model is proposed where gestures are modeled by HMMs and HMMs are combined to generate a garbage model.

Fang *et al.* proposed an alternative segmentation method, where instead of aligning the signs, transition parts are modeled using transition-movement models (TMMs) [5]. After dynamically clustering transition parts, an iterative segmentation algorithm is applied for automatically segmenting the continuous sentences. Sign classification is evaluated on a set containing 3000 sentence samples with a vocabulary of 5113 signs from Chinese Sign Language (CSL).

Vogler *et al.* discussed two approaches in order to solve the coarticulation problem [2]. In the first approach, tri-sign context dependent HMMs are trained analoguous to speech recognition systems. In the second approach, movement epenthesis is modeled explicitly. In order to cluster these, the k-means algorithm is used. The system's performance is shown on a continuous ASL database containing three-dimensional arm motions.

Nayak *et al.* presented an unsupervised approach to model phonemes (basic units of signs) from continuous sentences [14]. A histogram of pairwise distances of edge pixels is proposed for hand detection. Then DTW is applied to compute warped distance between two sub-sentences. A similar approach is used in a later study, where, in addition to segmentation, recognition performance is shown on several databases [15].

In Table 1.1 we list several gesture recognition systems in the literature, with

the dataset used, type of problem, and tracking, feature extraction, and classification methods employed.

1.3. Approach and Contributions

A good resource for signed Turkish is the TRT broadcast news for the hearing impaired. Aran et al. used broadcast news recordings for the hearing impaired for creating a sign dictionary [18]. The news data contains three information sources: sliding text, speech, and signs. In this system, the content of the video is extracted using the text and the speech information [43, 44] to generate segmented sign videos. The system enables a user to search for a word and retrieve the related sign videos from the data.

The tracking of the hands and the head is done by using a joint particle filter [45]. In this system the skin color probability images are needed for calculations. Therefore, first, the skin color pixels in the input image is determined using a trained Gaussian mixture model, and the resulting grey level image is smoothed by a Gaussian kernel to obtain skin probability image. Then, a joint particle filter is used to track a maximum of three objects: two hands and the face. In the end, the system represents each single object with a vector consisting of the position and the shape parameters. The shape parameters are selected as the width, the height and the angle of an ellipse surrounding the object.

The method is robust to occlusions of the hands and the face, and is able to recover fast if the tracking fails. The performance of the tracking module is shown on a 15 minute signing video for which the ground truth for the center of mass coordinates of the hand and face is manually annotated. It is shown that the tracking accuracy for the face is 99 per cent, whereas for the two hands it is 96 per cent.

In this thesis, we use a database containing videos recorded from TRT broadcast news for the hearing-impaired. Our aim is segmenting signs from these videos. The videos contain three information sources: speech, sliding text and signs. An example

Table 1.1. Some gesture recognition techniques.

Author(s)	**Dataset**	**isolated/ continuous**	**Hand Tracking**	**Feature Extraction**	**Classification**
Brand 1997 [8]	8 gestures	isolated & continuous	stereo camera	hand motion & hand shape	cHMM
Starner *et al.* 1998 [10]	40 ASL	continuous	skin color	hand motion & hand shape	HMM
Lee and Kim 1999 [11]	10 gestures	isolated & continuous	skin color	hand motion	HMM
Vogler and Metexas 1997 [2]	53 ASL	continuous	3 cameras & sensor system	hand motion & hand shape	HMM
Vogler and Metexas 1999 [3]	22 ASL	isolated	3D tracking system	hand motion	parallel HMM
Pavlovic 1999 [29]	11 gestures	isolated	skin color	hand motion & hand shape	HMM & cHMM & mixed-state DBN
Corradini 2001 [12]	5 gestures	isolated	skin color	body motion & shape	DTW
Fang and WenGao 2002 [4]	208 CSL	isolated & continuous	instrumented gloves	hand motion & hand shape	SRN & HMM
Liu *et al.* 2004 [13]	26 gestures	isolated	skin color	hand motion	HMM
Fang *et al.* 2007 [5]	5113 CSL	continuous	instrumented gloves	hand motion & hand shape	TMM
Farhadi and Forsyth 2007 [27]	31 ASL	continuous	skin color	hand motion & hand shape	HMM
Kong and Ranganath 2008 [6]	28 SEE	isolated	instrumented gloves	hand motion & hand shape	FLD and decision tree
Lichtenauer et.al 2008 [9]	120 DSL	isolated	stereo camera	hand motion & hand size	SDTW with CDFD & Q-DFFM
Nayak *et al.* 2008 [15]	147 ASL 7 gestures	isolated	skin color	distribution of edge orientations	DTW
Alon *et al.* 2008 [16]	10 gestures 27 ASL,	isolated & continuous	skin color	hand motion	DTW and HMM
Agris *et al.* 2008 [17]	263 BSL, 152 GSL,	isolated & continuous	skin color	hand motion and shape with facial features	HMM
Aran 2008 [19]	7 gestures 19 ASL	isolated	colored gloves	hand motion and shape with head motion	HMM

can be seen in Figure 1.1. These videos are segmented using the spoken term detection [43] and the segmentation results are given as input to our system. We also use the skin color probability images and the tracking results of the hands and the head [19].



Figure 1.1. An example frame from the news recordings. The three information sources are the speech, sliding text, and signs.

Our contribution in this thesis is to perform direct sign segmentation via sequence alignment techniques, without the need for a pre-trained sign model. We examine some feature extraction methods to accurately describe the motion and shape information of each hand. Then, we apply multiple sequence alignment techniques to find the common parts of several sequences. Here, we use the information obtained from the spoken term detection and assume that these sequences contain the same sign information. We also investigate recognition performances to compare feature extraction methods. The flowchart of the proposed system can be seen in Figure 1.2. Top part of the flowchart shows input to this work, as detailed in [43] and [45]. This thesis deals with preprocessing, feature extraction, and alignment of the videos.

1.4. Outline of the Thesis

First, we present a detailed analysis of different features of hand motion and shape, and compare their performances in Chapter 2.

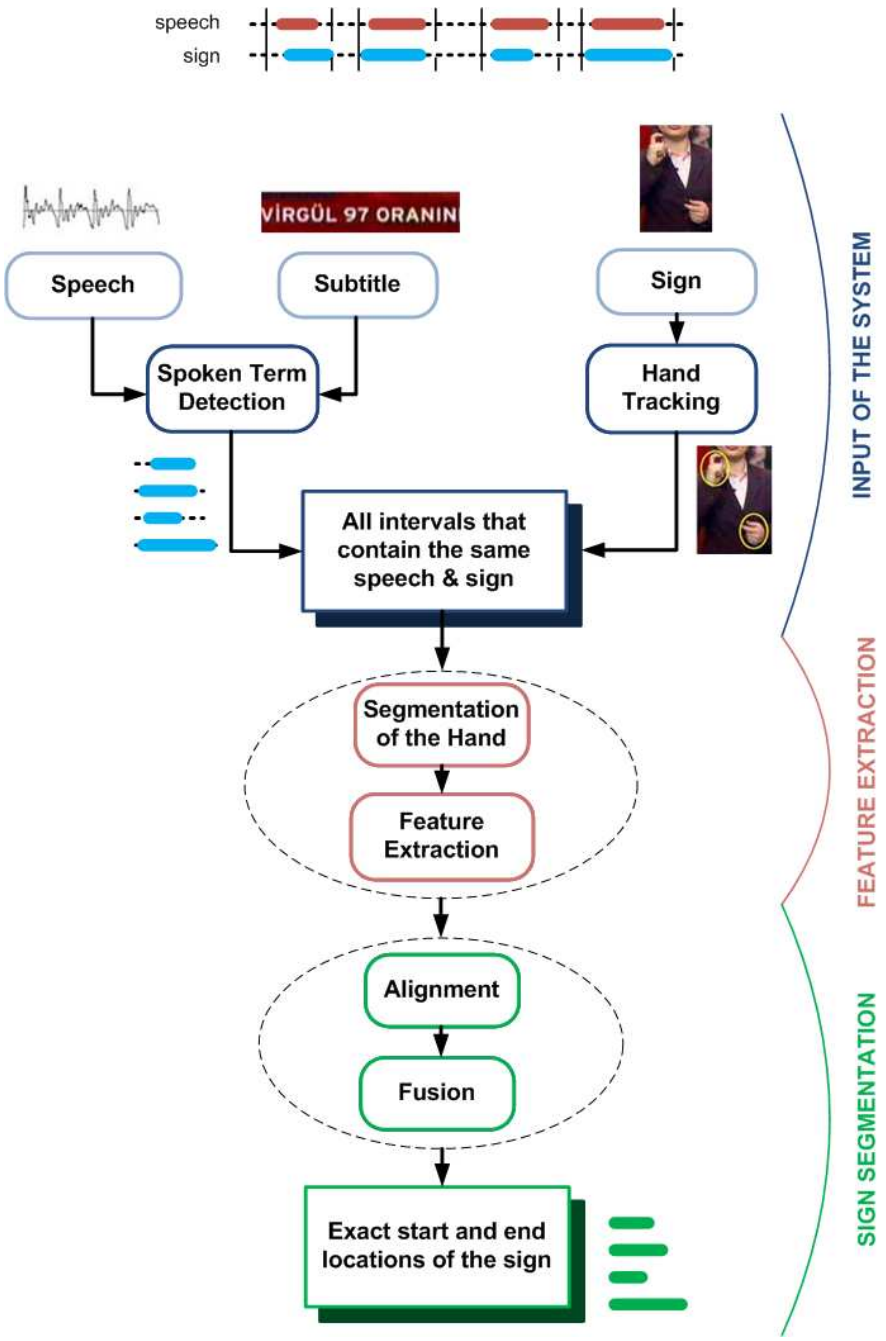


Figure 1.2. General System Flow.

In Chapter 3, the mathematical details of aligning the sign video sequences using Dynamic Time Waping, Continuous Hidden Markov Models and Coupled Hidden Markov Models are explained. First, we give a background for these methods and then explain our approach. The details of fusion of these methods conclude the Chapter 3.

Chapter 4 includes the experiments tested on a video database which is composed of Turkish signed speech videos and presents the achieved results. First, details of the database are given. Then, the experimental results of our alignment methods and recognition follows.

Finally, a summary of the results obtained is given with related discussions and future work in Chapter 5.

2. FEATURE EXTRACTION

Hand gestures are the basic components in sign language and usually they provide sufficient information to distinguish signs. In order to describe manual signals, efficient shape and motion features are needed. When 2D images are considered, appearance based approaches including region based methods and texture based methods are widely used in hand gesture recognition systems.

In this thesis, we compare several feature extraction methods in the scope of the continuous sign language recognition problem. We use 2D skin color hand images where resolution is low and occlusion of the hands with each other or with the face can be present. We extract eight sets of features as illustrated in Figure 2.1:

- **CoM**: Center of Mass coordinates of the hands;
- $\triangle$**CoM**: first order derivatives of Center of Mass coordinates;
- **E**: Ellipse parameters for each hand: major and minor axes, and rotation angle;
- **H**: Hu Moments;
- **RDF**: Radial Distance Function;
- **DCT**: Discrete Cosine Transform coefficients;
- **HOG**: Histogram of Oriented Gradients;
- **LBP**: Local Binary Patterns.

In this chapter, we first explain preprocessing modules which are responsible for segmentation of the hand image in the presence of noise or occlusion. Then, we provide details of our feature extraction algorithms. Comparative analysis of feature extraction methods will be given in Chapter 4.

2.1. Segmentation of the Hand

In this study, we are interested in continuous signs and there are no restrictions or assumptions on the dynamic shape and movement of the hand. The resolution of

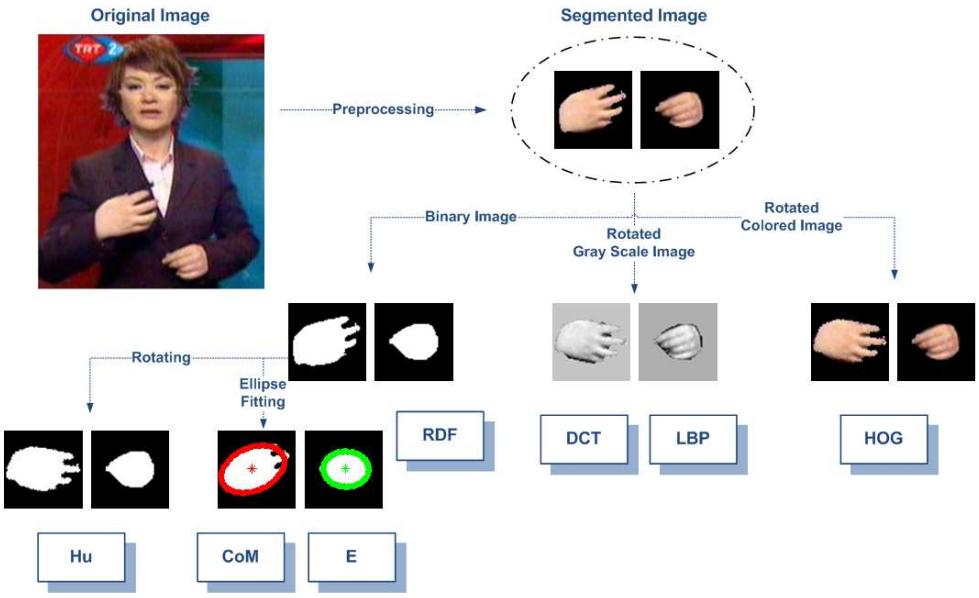


Figure 2.1. Feature extraction procedure.

the video is low. Especially when the hand is in motion, the hand images become blurred causing fingers not to be detected easily. Furthermore, the background is not homogenous: The signer stands beside a monitor where several videos are shown and the left hand may pass over this cluttered background. In addition, the signer's outfit changes in each video and the subtitle or the signer's outfit may resemble skin color. Moreover, the hand may occlude the head or the other hand. Consequently, segmentation of the hand can become a very challenging task.

In order to obtain reliable hand features, we first need to segment and normalize hand images. From the hand tracking module, we learn the probability image for the skin color regions, the center of mass coordinates, and the bounding ellipse parameters for each hand and the head. However, in some cases the bounding ellipse may not contain the whole hand. For example, the hand may be larger than the ellipse or when one finger is open, the ellipse may miss the finger. Therefore, we need some preprocessing to find a more elaborate segmented image containing the whole hand.

To obtain segmented hand images, we first apply region growing to the skin

colored regions enclosed by the ellipse which were found by the hand tracking module. The resulting images may contain some extra areas other than the hand region as a consequence of noise, occlusion or image characteristics. Therefore, after region growing, we check when the hands are occluded with each other or the person's neck, and apply template matching to correct the error in occluded images. Finally, we check when the area of the hands exceed a certain threshold to detect erroneous images. When head occlusion occurs, or inaccurate growth is detected, we discard the result of region growing and use the skin color regions enclosed by the initial ellipse as the outcome. The phases of preprocessing are illustrated in Figure 2.2.

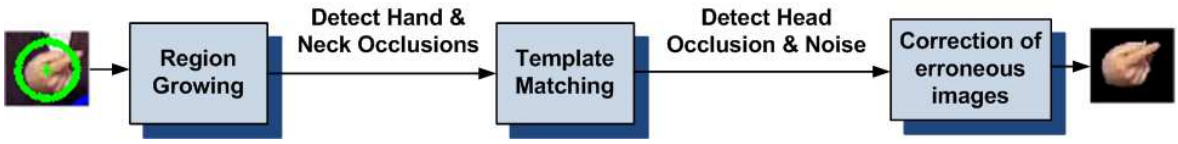


Figure 2.2. Phases of preprocessing.

2.1.1. Region Growing

In region growing, we aim to find the binary masks of hand regions in the image. For this purpose, we use the skin color probabilities and the ellipse parameters that we learn from the hand tracking module to obtain an initial region. Starting from this region, we apply region growing to the image.

In our algorithm, we first compute the mean color of the initial region and select all pixels at the 4-neighborhood of the boundary pixels as candidate pixels. Then we compute the Euclidean distances between the RGB values of the candidate pixels and the mean color. We find the candidate pixel that has the minimum distance and compare this distance to a predefined threshold to decide if the region growing should continue. To decide on the threshold value, we computed the standard deviations of the RGB values corresponding to the skin colored area on a small subset of images. Using the standard deviations, we computed the distances between the value of the farthest pixel to the mean. These values were observed to have a small variation around

80. Hence, in our computations we take the threshold value as 80. When the value is below this threshold, we add the pixel to the area, compute the new mean color, and find the new candidates. We stop the search when the area has grown by 60 per cent.

Region growing becomes very challenging when the hand is occluded with the face or the other hand. Therefore, we added an edge control to the region growing and prevent growing in the direction where an edge occurs. Here, we find the edges in the image using Canny edge detection [46] and store the indices of the edge pixels. Whenever we decide to add a new pixel to the growing area, we control if that pixel belongs to an edge. If this is the case, we delete this pixel from the list of neighbors.

Despite the edge control, the resulting segmentation may include some skin color regions that do not belong to the hand. Consequently, in order to detect the occurrence of such occlusions, we apply segmentation to the head and neck regions which do not need to be accurately found. For the head, we accept the skin colored area bounded by the initial ellipse as the actual head area. We do not have any segmentation information on the neck, thereby we expand the head region to twice its initial size. In the end, we compute the difference between the enlarged head region and the actual head region to define a region containing the neck. When the hand enters the head or neck area, we decide that occlusion has occurred and apply template matching to improve segmentation.

2.1.2. Template Matching

When occlusion occurs we use template matching to find the actual size, position and orientation of the hand by comparing the occluded image with an image where segmentation is successful. One approach is to match the occluded image to the previous frame in the video sequence but in this case, the mistakes accumulate. Therefore, we match the occluded image to the last non-occluded image.

In the algorithm we use the occluded image as the template and the non-occluded image as the target. First, we subtract the mean from the images to eliminate illumi-

nation differences. Then, we rotate the target image using the angles in an interval of $[-10, 10]$ degrees, and translate the target image to all possible locations in the template image. Next, we compute the correlations between the template image and the transformed target image using the Equation 2.1.

$$\mathbf{C_{I_{temp},I_{tar}}} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{M}(I_{temp}(x,y)I_{tar}(x,y))}{\sqrt{\sum_{x=1}^{N}\sum_{y=1}^{M}I_{tar}(x,y)}} \tag{2.1}$$

where, $I(x, y)$ is the intensity value of the pixel at the coordinates (x, y) and N, M are the height and width of the target image, respectively.

In the end, we find the rotation and translation values to obtain an image with the highest correlation to the template image and use the transformed non-occluded image as the segmentation result.

2.1.3. Elimination of Poor Segmentation

We observed that using template matching we obtain acceptable segmentation results when only a partial occlusion between the hand and the face occurs or when the hand occludes an almost uniform skin colored region. However, when the hand is in full occlusion with the face, the complex pattern of the face and the long duration of the occlusion results in poor segmentation. Moreover, we cannot previously detect when a non-skin colored area is included to the hand region. Hence, to eliminate poor segmentation, further precautions must be taken.

In order to find the images with poor segmentation, we compute area differences in successive hand images and control if the hand's size has increased by reasonable amounts. Here, we compare the differences to a predefined threshold and if the region is larger than this threshold we decide that this image includes some non-hand regions.

When the segmentation is successful, the hand region is approximately 3000 pixels of size. Therefore, in our computations we take the threshold as 200 pixels. When we detect that the segmentation is poor, we accept the skin color area bounded by the initial ellipse that has been found for the hand as the actual area. The results can be seen in Figure 2.3.

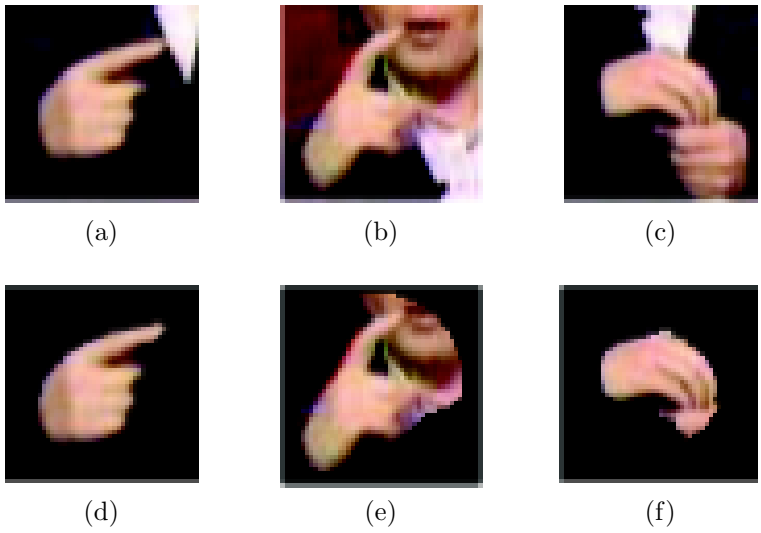


Figure 2.3. Output of segmentation. (a) Non-occluded image, (b) face occlusion, (c) occlusion with the other hand, and (d), (e) and (f) are the corresponding results for segmentation.

2.2. Center of Mass (CoM)

The Center of Mass (CoM) coordinates are used to describe the trajectory and the speed of the hands. To find the CoM coordinates, we calculate the mean of the coordinates of the pixels belonging to the corresponding area obtained in the segmentation process. For describing speed, we calculate first order derivatives of the CoM coordinates ($\triangle$CoM) for left and right hands. The resulting CoM positions can be seen in Figure 2.4.

2.3. Ellipse (E)

As a simple shape feature to represent the hand shape, we fit an ellipse to the hand images and calculate the ellipse parameters: major and minor axes and the rotation angle. These simple features do not contain any information regarding the hand contour. Instead, they give a rough idea about the shape and orientation of the hand. A representation of the Ellipse features can be seen in Figure 2.4.

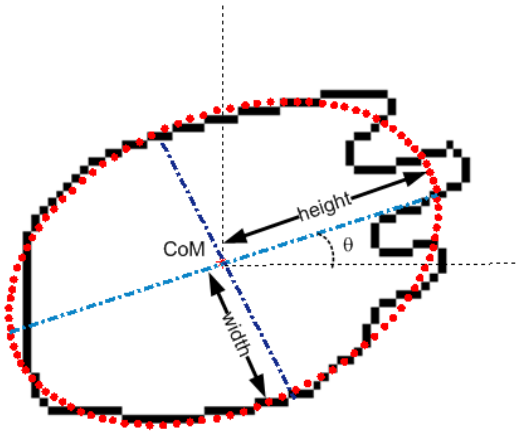
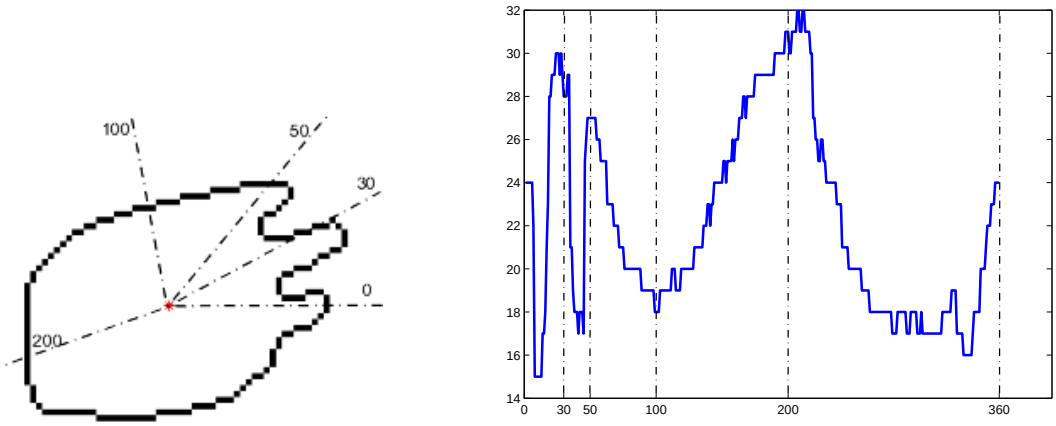


Figure 2.4. A representation of the CoM and Ellipse parameters which are orientation angle, height and width

2.4. Radial Distance Function (RDF)

Radial Distance Function (RDF) is a method to describe the outer contour of an image. A reference point inside the closed contour is chosen and the distance of this reference point to the curve as a function of angle is plotted as seen in Figure 2.5.

In hand gesture recognition, when fingers are visible and separated, RDF can be used to detect and localize the fingertips [47]. In our computations, we take the CoM coordinate as the reference point and compute RDF for every five degrees.



(a) (b)

Figure 2.5. RDF configuration procedure.

2.5. HU Moments

Image moments are used in several computer vision applications for representing global and invariant shape characteristics of image features. Hu stated that the continuous two-dimensional $(p+q)^{th}$ order moments of a density distribution function $\rho(x,y)$ are defined in terms of Riemann integrals [48] as:

$$m_{pq} = \iint \rho(x,y) x^p y^q dxdy \tag{2.2}$$

where $p,q = 1,2,\ldots,\infty$. When we consider an image consisting of pixels as a two-dimensional discrete Cartesian density distribution, image moments of order $(p+q)$ are computed as:

$$m_{pq} = \sum_{x=1}^{N} \sum_{y=1}^{M} I(x,y) x^p y^q dxdy \tag{2.3}$$

where N, M are the image dimensions and x, y are the pixel coordinates. In order to achieve invariance under translation, centralized moments are computed as in Equa-

tion 2.4, and further normalized for eliminating the scale factor as in Equation 2.5.

$$m_{pq}^{C} = \sum_{x=1}^{N}\sum_{y=1}^{M} I(x,y)(x-\bar{x})^{p}(y-\bar{y})^{q} \tag{2.4}$$

$$\eta_{pq} = \frac{m_{pq}}{m_{00}^{\frac{p+q}{2}+1}} \tag{2.5}$$

Using the normalized central moments up to order three, Hu derived a set of rotation invariant moments which are referred to as Hu moments [48]. The first seven of these moments are frequently used for scale, translation, and rotation invariant pattern identification. The first six are also reflection invariant whereas the seventh moment is skew orthogonal invariant, which is useful in distinguishing mirror images. The Hu moment feature vector $\mathbf{v_{HU}}$ is calculated as in Equations 2.6-2.12.

$$\mathbf{v_{HU}}(1) = \eta_{20}+\eta_{02} \tag{2.6}$$

$$\mathbf{v_{HU}}(2) = (\eta_{20}+\eta_{02})^2+4\eta_{11}^2 \tag{2.7}$$

$$\mathbf{v_{HU}}(3) = (\eta_{30}-3\eta_{12})^2+(\eta_{03}-3\eta_{21})^2 \tag{2.8}$$

$$\mathbf{v_{HU}}(4) = (\eta_{30}+\eta_{12})^2+(\eta_{03}+\eta_{21})^2 \tag{2.9}$$

$$\begin{aligned}\mathbf{v_{HU}}(5) = {} & (\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})\big((\eta_{30}+\eta_{12})^2-3(\eta_{03}+\eta_{21})^2\big) \\ & -(\eta_{03}-3\eta_{21})(\eta_{03}+\eta_{2}1)\big(3(\eta_{30}+\eta_{12})^2-(\eta_{03}+\eta_{21})^2\big)\end{aligned} \tag{2.10}$$

$$\begin{aligned}\mathbf{v_{HU}}(6) = {} & (\eta_{20}-\eta_{02})\big((\eta_{30}+\eta_{12})^2-(\eta_{03}+\eta_{21})^2\big) \\ & +4\eta_{11}^2(\eta_{30}+\eta_{12})(\eta_{03}+\eta_{21})\end{aligned} \tag{2.11}$$

$$\begin{aligned}\mathbf{v_{HU}}(7) = {} & (3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})\big((\eta_{30}+\eta_{12})^2-3(\eta_{03}+\eta_{21})^2\big) \\ & -(\eta_{30}-3\eta_{12})(\eta_{03}+\eta_{21})\big(3(\eta_{30}+\eta_{12})^2-(\eta_{03}+\eta_{21})^2\big)\end{aligned} \tag{2.12}$$

In our computations, we compute the Hu moments from the binary mask of the segmented hand. Due to segmentation error in the presence of occlusion, the segmented image may be larger than the actual hand area. Therefore, we crop the image to a 64×64 sized window with the hand in the center. Thus, the hand is included in the window whereas some non-manual parts are excluded. We observed a small improvement in the performance when we included the rotation information to our feature

vector. Therefore, prior to calculating the Hu moments, we rotate the images using the rotation angle that we obtained from the computation of ellipse parameters, and we attach the rotation angle to our final feature vector.

2.6. Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) is a well-known signal analysis method which is frequently used in data representation and classification studies such as face recognition [49] and hand gesture recognition [26]. DCT expresses the data points in terms of a sum of cosine functions as in Equation 2.13.

$$\mathbf{V_{DCT}}(u,v) = \varphi(u)\varphi(v)\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} p(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2M}\right] \tag{2.13}$$

$$\varphi(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for} \quad u = 0 \\ \sqrt{\frac{2}{N}} & \text{for} \quad u = 1,2,\ldots,N-1 \end{cases} \tag{2.14}$$

where, $x, y, u, v = 0, 1, \ldots, N-1$, $I(x,y)$ is the intensity value at (x,y), N, M are the image dimensions and x, y are the pixel coordinates.

When applied to a two-dimensional image, DCT converts the spatial representation in the image into a frequency representation in a 2D matrix. The upper left corner of this resulting matrix contains the low frequency components, which contain most of the information in the image. Thus, one can discard the high frequency coefficients to reduce dimension of the feature vectors. The element in upper left corner of the matrix corresponding to the $(0,0)$ component represents the average intensity value of the image and is referred to as the DC coefficient. The $(0,1)$ and $(1,0)$ components represent the average vertical and horizontal intensity changes, respectively.

In our method, we first fill the background with the mean color of the segmented area and convert the segmented hand image to gray scale. Then, we crop the image to

obtain images of size 64×64 pixels where the hand is located in the center. Before we calculate the DCT coefficients, we rotate the hands using the rotation angle obtained by the computation of ellipse parameters to obtain scale invariant descriptors. Then, we divide each hand image into blocks of size 8×8 pixels size. On each block we apply DCT and order DCT coefficients using zig-zag scanning as in Figure 2.6.

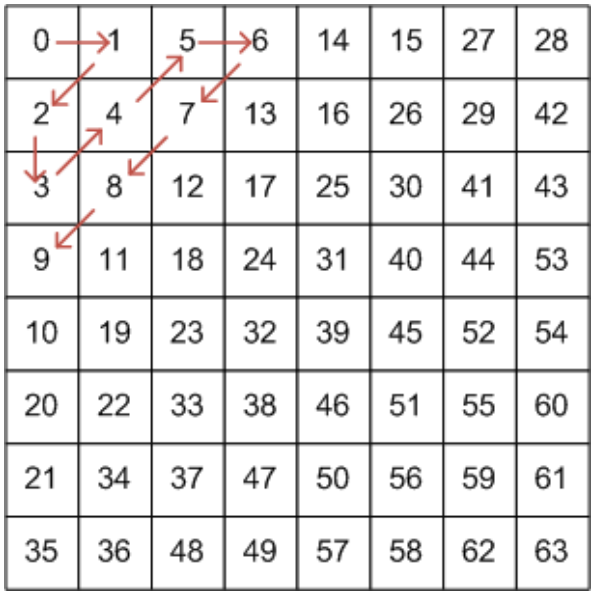


Figure 2.6. The procedure of zig-zag scanning.

As most of the information in DCT is concentrated in the lower frequencies, we concentrate on these frequencies. In our computations we eliminate the DC coefficient, since it can be directly affected by illumination variations. Then we select the first five DCT coefficients from the remaining ones in each of the 64 blocks. To construct the feature vector we concatenate the DCT coefficients obtained from each block and obtain vectors with dimension of 320 to represent each image.

For feature normalization, we follow a similar method as in [49]. To eliminate the effect of illumination changes in each block, we first normalize the total magnitude of each block's DCT coefficients to unit norm using Equation 2.15.

$$\mathbf{v_{DCT}}^b = \frac{\mathbf{v_{DCT}}^b}{\|\mathbf{v_{DCT}}^b\|} \tag{2.15}$$

where $\mathbf{v_{DCT}}^b$ represents the DCT coefficient vector of the bth block. To balance the effect of each DCT coefficient, we divide the coefficients to the standard deviations learned from a training set as in Equation 2.16.

$$\mathbf{v_{DCT}}(i) = \frac{\mathbf{v_{DCT}}(i)}{\sigma(\mathbf{v_{DCT}}(i))} \tag{2.16}$$

where $\mathbf{v_{DCT}}(i)$ represents the ith DCT coefficient, and $\sigma(\mathbf{v_{DCT}}(i))$ represents the standard deviation of the ith coefficients. We use leave-one-out cross validation. In each iteration we put one sample to the test set and the remaining samples to the training set. The procedure to obtain DCT features can be seen in Figure 2.7.

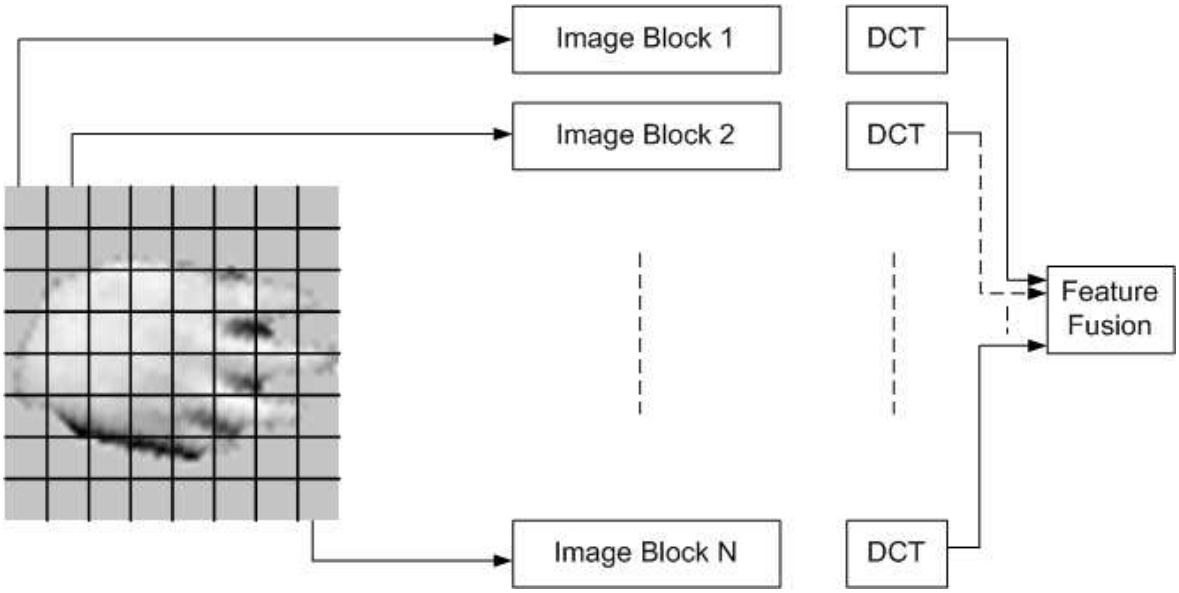


Figure 2.7. The procedure to obtain DCT features.

2.7. Histogram of Oriented Gradients (HOG)

HOG descriptors are used in computer vision as feature descriptors in object detection [50] and recognition applications [25]. This method is based on the idea that shape within an image can be described by the distribution of local intensity gradients or edge directions. Hence, it counts occurrences of gradient orientation in localized portions of an image. The advantage of using HOG descriptors is that they can capture characteristic edge or gradient structure and offer some robustness to scene

illumination changes.

In our method, as a preprocessing step, we rotate the hand using the rotation angle and translate it to the center of a 64×64 sized box. To compute the gradient image, we apply the centered 1D point derivative, which is the mask $[-1, 0, 1]$, for each color channel both in the vertical and horizontal directions to find g_V and g_H. Then, we find the gradient magnitude $\|g\|$, and gradient orientation Θ of each pixel using Equations 2.17 and 2.18.

$$\|g\| = \sqrt{(g_V^2 + g_H^2)} \tag{2.17}$$

$$\Theta = \arctan\left(\frac{g_V}{g_H}\right) \tag{2.18}$$

We take the gradient magnitude with the largest norm and the corresponding gradient orientation as the pixel's gradient values. Then, we divide the image into non-overlapping cells with 8×8 pixels size. For each cell, we compute an orientation histogram having nine bins. The orientation bins are evenly spaced between $[-\pi/2, \pi/2]$. Each pixel in the cell calculates a weighted vote for the histogram based on the gradient orientations. To obtain the weights, we multiply the gradient magnitude matrix of each cell with the gaussian window which is computed as in Equation 2.20 and use the outcome as the weight of the corresponding pixel;

$$\omega(x, y) = \omega(x)\omega(y) \tag{2.19}$$

$$\omega(n) = e^{-\frac{1}{2}\left[\frac{n-(L-1)/2}{0.4(L-1)/2}\right]^2} \tag{2.20}$$

where L is the window length, which in our case is the cell size.

To eliminate gradient variations resulting from the local illumination changes, we perform an overlapping local contrast normalization approach as in [50]. Here, we group cells into overlapping blocks of 16×16 pixels size, where each block contains four cells with nine dimensional feature vectors. We concatenate these vectors to obtain 36

dimensional feature vectors for each block and normalize using Equation 2.21,

$$\mathbf{v_{HOG}}^{b} = \frac{\mathbf{v_{HOG}}^{b}}{\sqrt{\|\mathbf{v_{HOG}}^{b}\|_2 + \varepsilon^2}} \tag{2.21}$$

where $\mathbf{v_{HOG}}^{b}$ is the HOG coefficient vector of the bth block, ε is a small constant of size 10^{-10} and $\|.\|_2$ represents Euclidean norm. The resulting images can be seen in Figure 2.8.

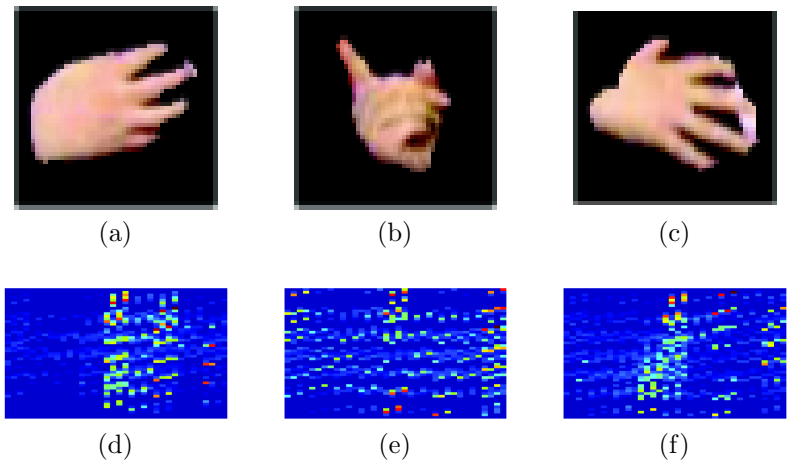


Figure 2.8. Example outputs of the HOG method

2.8. Local Binary Patterns (LBP)

The local binary pattern (LBP) representation is used for describing local spatial structure of an image. The most important property of the LBP operator is its invariance against illumination changes. Moreover, its computational simplicity makes it possible to analyze images in challenging real-time settings. The LBP method is widely used in studies such as face recognition [22] and facial expression recognition [23].

LBP of a given pixel is computed by comparing its intensity value with the intensities of the 8-neighborhood pixels, where the value of the center pixel is used to threshold the neighborhood. If the neighboring pixel value is greater than or equal to the center pixel value, then this pixel is labeled as one, otherwise zero. The ordered set of the binary comparisons form an LBP code, and the decimal value of this binary

code is used to represent the local structural information around the given pixel. An example of computing LBP in a 3×3 neighborhood is shown in Figure 2.9.

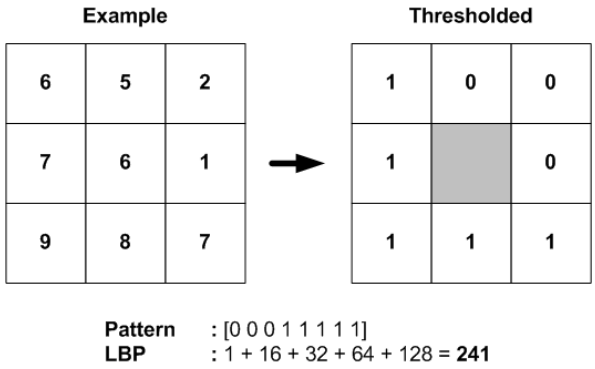


Figure 2.9. An example of computing LBP in a 3×3 neighborhood

Instead of using and the 256-bin histogram of the LBP labels as a texture descriptor, it is possible to use only the uniform patterns without losing much information. An LBP pattern is a uniform pattern if it contains at most two bitwise transitions from zero to one or vice versa when the binary string is considered circular. For example, 00000000, 00111000, and 11100001 are uniform patterns, whereas 11110101 is a non-uniform pattern. There are totally 58 different uniform patterns at eight-bit LBP representation as in Figure 2.10. When we assign the remaining patterns in one non-uniform binary number we can represent the texture structure with a 59-bin histogram.

In our computations, we first transform the gray-scale image to the LBP domain. The resulting images can be seen in Figure 2.11. The obtained LBP-image is then normalized and divided into non-overlapping blocks of 8×8 pixels resolution. On each block, we apply DCT.

2.9. Postprocessing of the Features

In our database, hand positions and size can be noisy due to segmentation error, and they depend on the signer's position in the scene which results in translation and

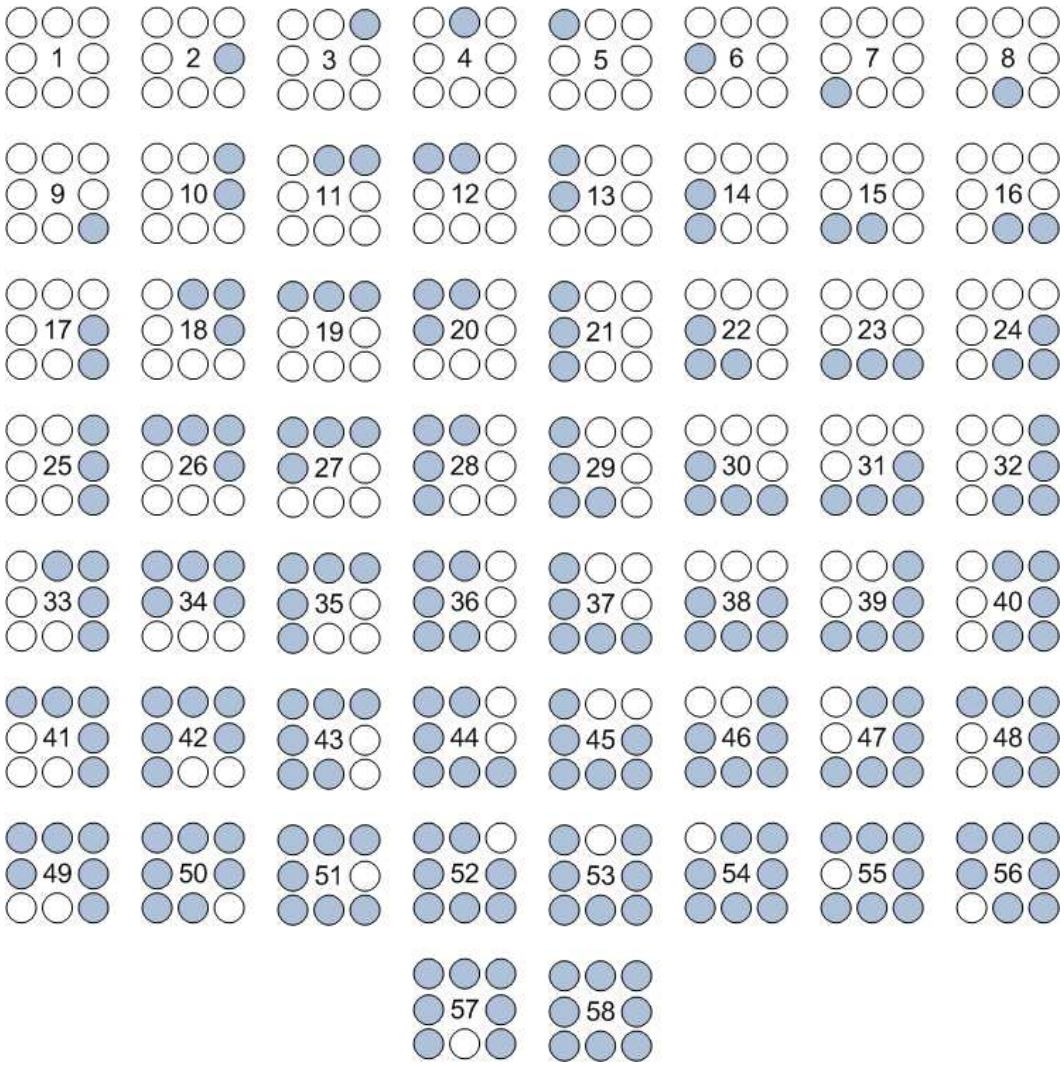
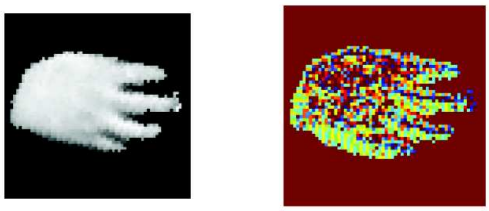


Figure 2.10. Uniform patterns encoded in LBP.



(a) (b)

Figure 2.11. A sample output of the LBP method.

scale variance. The feature sequence may contain some gaps when the hand disappears or the tracking algorithm fails to detect the hand. Moreover, some feature vectors have a large dimensionality which complicates the storage and computation. Therefore, after extracting features, we apply some postprocessing to achieve efficiency.

To obtain translation invariance, we take the face as the center of our coordinate system and recalculate the CoM coordinates of the hands accordingly. Gaps occur when the signer makes a pause and the hands disappear, or when the hand tracking algorithm cannot detect the hands. In the former case, hands appear and disappear from nearby locations and the hand shape during the gap does not contain any information. In the second case, hand tracking algorithm is able to recover fast if the tracking fails, and location and shape of the hand changes only slightly due to short duration of the gap. Hence, we fill the gaps in ellipse and CoM features using linear interpolation, whereas for the other features we accept the feature of the last detected hand as unchanged during the gap. Then, we apply a moving average filter to the features along the trajectory of the hands to eliminate noise. The algorithm is given in Figure 2.12.

for $f = 1$ to F **do**
  if f=1 or f=F **then**
    $\mathbf{v}^f(i) \Leftarrow \mathbf{v}^f(i)$
  end if
  if $f = 2$ or $f = F - 1$ **then**
    $\mathbf{v}^f(i) \Leftarrow (\mathbf{v}^{f-1}(i) + \mathbf{v}^f(i) + \mathbf{v}^{f+1}(i))/3$
  end if
  if $f < 2$ and $f < F - 1$ **then**
    $\mathbf{v}^f(i) \Leftarrow (\mathbf{v}^{f-2}(i) + \mathbf{v}^{f-1}(i) + \mathbf{v}^f(i) + \mathbf{v}^{f+1}(i) + \mathbf{v}^{f+2}(i))/5$
  end if
end for

Figure 2.12. Moving average filter algorithm.

Finally, we apply min-max normalization to each video sequence using Equa-

tion 2.22 to obtain scale invariance.

$$\mathbf{v}^f(i) = \frac{\mathbf{v}^f(i) - \min_f(\mathbf{v}^f(i))}{\max_f(\mathbf{v}^f(i)) - \min_f(\mathbf{v}^f(i))} \tag{2.22}$$

where $\mathbf{v}^f(i)$ is the ith component of the feature vector at frame f.

When the dimension of the feature vector is high as in HOG, DCT, RDF and LBP features, we apply Principal Component Analysis [51] (PCA) to reduce the dimensionality. PCA is an unsupervised method to find an orthogonal linear transform to a new coordinate system where the variance is maximized. PCA algorithm finds the orthonormal vectors that span this new coordinate system. The principal components are found by projecting the original data using these vectors, where these components are sorted according to their ability to express the variance in the data. The proportion of variance explained by the first k principal components (Λ^k) is computed as in Equation 2.23:

$$\Lambda^k = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d} \tag{2.23}$$

where, λ_i are the eigenvalues sorted in descending order. When the dimension is reduced to a value which explains more than 90 per cent of the variance, we can assume that the loss of information is in acceptable amounts.

For computing PCA parameters for a given sign, we perform leave-one-out cross validation. At each instance we perform training with one example in the test set and the remaining examples in the training set. We reduce the dimension so that 90 percent of the variance is explained. After reducing the dimension, we normalize the feature components.

Finally, we concatenate the feature vectors belonging to the right hand and the left hand to obtain the final feature vector. Moreover, when we rotate the image before applying the feature extraction method as in LBP, H, HOG and DCT, we add the

rotation angle to the feature vector since the information of the hand orientation is lost.

3. ALIGNMENT TECHNIQUES

Alignment is a way of synchronizing different time series, and is used to identify regions of similarity in multiple sequences. When the sequences coincide and the timing differences between them are eliminated, analogous segments can be extracted using segmentation.

In our study, we aim to segment Turkish signed speech video sequences using multiple sequence alignment techniques to obtain distinct sign videos. In these videos there is an accompanying speech information which performs a simultaneous translation to the signs. The video sequences are segmented using the spoken term detection [43] and the segmentation results give a rough interval for the signs. Our aim is to find the exact interval of the signs via sequence alignment techniques.

In this chapter, we explore two alignment techniques and their variants: Dynamic Time Warping (DTW) and Hidden Markov Models (HMM). After we explain computation details of the algorithms, we show how to use them for the task of segmentation. Moreover we introduce some fusion techniques to improve our performance of segmentation.

3.1. Dynamic Time Warping (DTW)

DTW is a widely used alignment approach in the field of bioinformatics [31, 32], online or off-line signature verification [33, 34], and speech recognition [35]. Generally, it is used for pairwise alignment since the extension of the problem is NP-complete. However, several multiple alignment techniques are proposed which exploit pairwise alignment using DTW [32].

The first step in DTW to align two sequences is to calculate the local score matrix of the sequences. Suppose we have two sequences $\mathbf{X}$ and $\mathbf{Y}$ of length F_1 and

F_2 respectively, as in Equations 3.1 and 3.2.

$$\mathbf{X} = \mathbf{v}_X^1, \mathbf{v}_X^2, \ldots, \mathbf{v}_X^i, \ldots, \mathbf{v}_X^{F_1} \tag{3.1}$$

$$\mathbf{Y} = \mathbf{v}_Y^1, \mathbf{v}_Y^2, \ldots, \mathbf{v}_Y^j, \ldots, \mathbf{v}_Y^{F_2} \tag{3.2}$$

where, $\mathbf{v}_X^i$ is the ith value or vector of the sequence $\mathbf{X}$, and y_Y^j is the jth value or vector of the sequence $\mathbf{Y}$. Then, the local score matrix is an $F_1 \times F_2$ matrix where the element $\mathbf{D_L}(i,j)$ of the matrix contains the distance between the two components $\mathbf{v}_X^i$ and $\mathbf{v}_Y^j$. Once the score matrix is calculated, it is aimed to find the alignment path that satisfies the following conditions:

- Boundary conditions: The alignment path must start and end in diagonally opposite corner cells of the matrix.
- Continuity: The allowed steps in the alignment path is restricted to adjacent cells.
- Monotonicity: The alignment must be monotonically spaced in time.

To satisfy these conditions, an accumulated distance matrix $\mathbf{D_A}$ is constructed using the algorithm shown in Figure 3.1.

In the final step, the accumulated distance matrix is traced backwards to obtain the alignment path. In back-tracing, the path is constructed starting from (F_1, F_2) and iteratively choosing the adjacent coordinate having the minimum accumulated distance value and satisfying the monotonicity condition. This is illustrated in Figure 3.2.

In our study, we perform the multiple alignment of the sequences via pairwise alignments for each sign. First we construct the local score matrix where we use Euclidean distance as local distance. Once the score matrix is calculated we need to construct the accumulated distance matrix to find the alignment path.

Since we aim to align video sequences of continuous sign language, there is a high possibility of having junk frames unrelated to the sign at the start and end of

Construct the score matrix S_L

for $i = 1$ to F_1 **do**

  for $j = 1$ to F_2 **do**

    if $i = 1$ and $j = 1$ **then**

      $\mathbf{D_A}(i, j) = \mathbf{D_L}(i, j)$;

    end if

    if $i = 1$ and $j > 1$ **then**

      $\mathbf{D_A}(i, j) = \mathbf{D_L}(i, j) + \mathbf{D_A}(i, j - 1)$;

    end if

    if $i > 1$ and $j = 1$ **then**

      $\mathbf{D_A}(i, j) = \mathbf{D_L}(i, j) + \mathbf{D_A}(i - 1, j)$;

    end if

    if $i > 1$ and $j > 1$ **then**

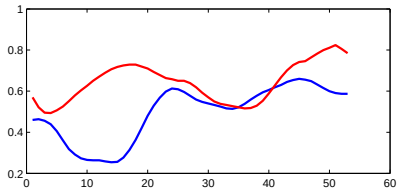
      $\mathbf{D_A}(i, j) = \mathbf{D_L}(i, j) + \min(\mathbf{D_A}(i - 1, j), \mathbf{D_A}(i, j - 1), \mathbf{D_A}(i - 1, j - 1))$;

    end if

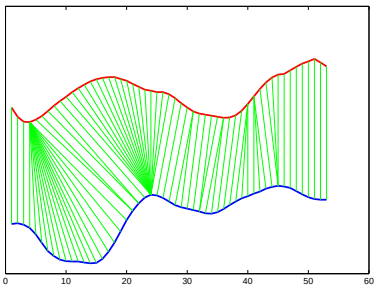
  end for

end for

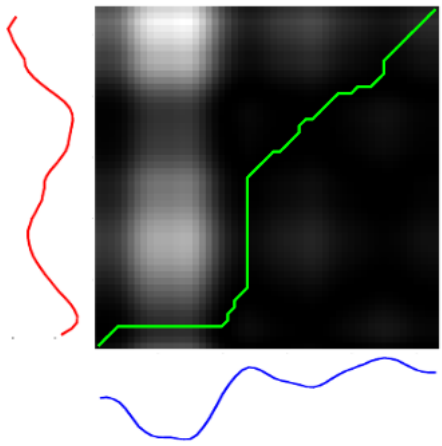
Figure 3.1. Construction of accumulated distance matrix.



(a)



(b)



(c)

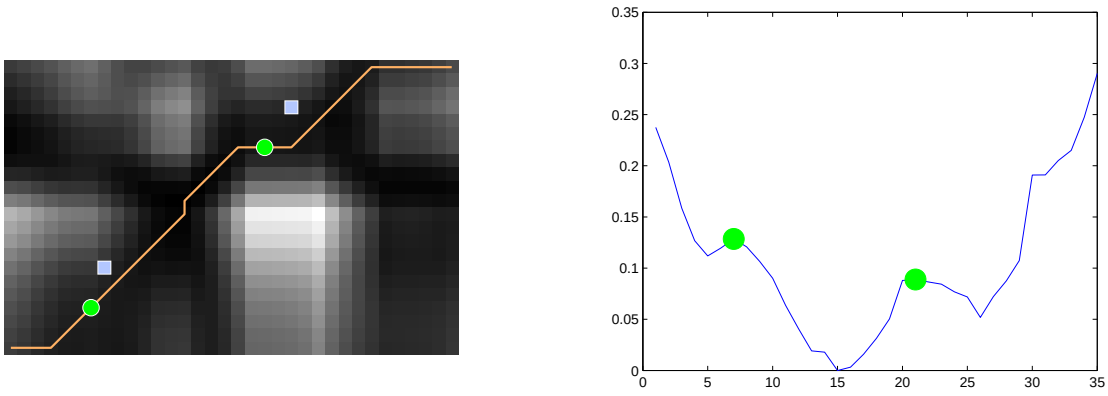
Figure 3.2. Alignment with DTW. We align two sequences that correspond to right hand movements of the sign "prime minister". (a) The change in horizontal positions of the hands in time, (b) the resulting alignment, and (c) the local score matrix and the alignment path. Note that the squares represent the local distances and as the distance decreases, the color gets darker

the sequences. Therefore, despite the general approach, we construct two accumulated distance matrices starting from an interior location and ending in diagonally opposite corner cells. To determine the starting location of these matrices, we search the window that is located in the center of the local score matrix and has one third the dimensions of this matrix. We select the point with the lowest score inside this window as our starting point. Then, we compute the alignment paths according to these matrices and concatenate them to obtain the final alignment. This approach increases the possibility of starting from a point which belongs to the optimal alignment.

Using pairwise alignment, we align all samples of a sign in all possible combinations and obtain corresponding alignment paths. Then we consider the local distances along these paths. When the alignment is successful, local distances must be small. Therefore, we can assume that local distances will start decreasing when the searched sign starts, and start increasing when the sign ends. Hence, the start and end locations correspond to local maxima points of the local scores along the alignment path. Moreover, we want to find a segment that has minimum total distance and maximum length. To ensure both of these requirements, we find the local maxima locations and compute the dynamic time warping score S for all possible intervals using:

$$S = \frac{\sqrt{\sum_{x,y \in P_W|_{l_m}^{l_n}} \mathbf{D_L}^2(x,y)}}{\sum_{x,y \in P_W|_{l_m}^{l_n}} 1} \tag{3.3}$$

where $\mathbf{D_L}$ is the local score matrix, l_m and l_n are the mth and n^{th} local maxima locations, and $P_W|_{l_m}^{l_n}$ is the alignment path interval between the local maxima locations l_m and l_n. We choose the interval with the minimum score as the desirable segment. An example for segmentation using DTW is shown in Figure 3.3. Consequently, we obtain several candidate start and end locations from each alignment. We determine the actual locations by averaging these candidate locations.



(a) (b)

Figure 3.3. Segmentation with DTW. (a) The local score matrix of two videos and the alignment path, and (b) the local scores along the alignment path. Green spots show the found locations, blue spots show the ground truth locations.

3.2. Hidden Markov Models (HMM)

Hidden Markov models (HMM) are especially known for their application in temporal pattern recognition tasks such as speech [43], handwriting [52], gesture recognition [29] and bioinformatics [32]. They are usually preferred for their ability to provide an efficient way of dealing with temporal variability among sequences and missing data [53].

HMMs are statistical models which are assumed to be Markov processes where the states are hidden. Each time a state is visited, an observation is generated with a probability conditioned on that state. The objective is to determine the hidden parameters using the recorded observation sequences where the conditional probability density of an observation at a given time depends only on the most recent past observations.

In this study we use left-to-right continuous HMMs and coupled HMMs to model each sign. Using these models, we find the state sequences of each sign to represent the alignment. This section will discuss the details of modeling gestures with left-to-right

HMMs and coupled HMMs.

3.2.1. Left-Right Hidden Markov Models

An HMM having Q states assumes that the observation strings can be divided into Q units, where each observation in a unit is generated according to a probabilistic function of the corresponding state. To explain a number of observation sequences, an HMM is modeled consisting of the model parameters: initial state probabilities $\mathbf{\Pi}$, state transition probabilities $\mathbf{A}$, and observation probabilities $\mathbf{B}$.

In a fully connected HMM, each state has a probability of making a transition to itself or others states. The transition probability from state i to state j can be given as:

$$a_{ij} \equiv P(q_{t+1} = s_j | q_t = s_i), \qquad i, j = 1, \ldots, Q \tag{3.4}$$

where q_t is the state variable and s_i is the probability of being in state i at time t. These values form a $Q \times Q$ matrix $\mathbf{A}$ with elements a_{ij} satisfying:

$$a_{ij} \geq 0 \quad \forall i, j \tag{3.5}$$

$$\sum_{j=1}^{Q} a_{ij} = 1 \tag{3.6}$$

In left-to-right HMMs, the transition to other states is restricted so that the states are ordered in time and transition is only allowed when it is made to a state with an index that is greater than or equal to the index of the current state (See Figure 3.4).

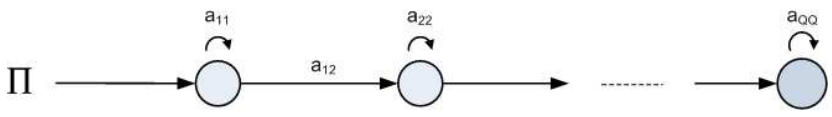


Figure 3.4. Left-to-right HMM.

To explain how the first state is chosen, initial state probabilities $\mathbf{\Pi}$ are given where the initial probability π_i for each state i is defined as:

$$\pi_i \equiv P(q_1 = s_i), \qquad i = 1, \ldots, Q \tag{3.7}$$

having the property

$$\sum_{i=1}^{Q} \pi_i = 1 \tag{3.8}$$

The observation string is composed of T elements which can be discrete or real valued. In the case of discrete observations, the elements of the sequence are symbols of an alphabet $\mathbb{V}$ with M distinct symbols:

$$\mathbb{V} = \{v_1, v_2, \ldots, v_M\} \tag{3.9}$$

and the probability of observing the symbol v_m in state j is given as:

$$b_j(m) \equiv P(O_t = v_m | q_t = s_j) \tag{3.10}$$

where O_t is the observation at time t. Similarly, these values form a $Q \times M$ matrix $\mathbf{B}$ satisfying:

$$\sum_{m=1}^{M} b_j(m) = 1 \tag{3.11}$$

In the continuous case, observation sequences are composed of F real valued vectors of dimension d. One possibility is to convert them to discrete values using vector quantization. Another possibility is to use multivariate Gaussian distributions to model the probability function which is used to find observation probabilities. Here, for each state i having mean μ_i and covariance Σ_i, the probability of observing the

vector $\mathbf{v}$ of dimension d is computed as:

$$b_i(\mathbf{v}) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{v}-\mu_i)^T\Sigma_i^{-1}(\mathbf{v}-\mu_i)\right) \tag{3.12}$$

The model parameters $\lambda = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ of an HMM can be learned using an Expectation Maximization (EM) procedure called the Baum-Welch algorithm. EM procedure iteratively calls two steps, the E-step and the M-step. In the beginning, the model parameters $(\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ are initialized randomly. Then, in the E-step two variables γ and ξ are computed, given the current model λ; and in the M-step, λ is recalculated given γ and ξ. These steps are repeated until the likelihood converges.

There is an efficient procedure to calculate $P(O|\lambda)$, which is called the forward-backward procedure. Here the forward variable $\alpha_t(i)$ is defined as the probability of observing the partial sequence $\{O_1...O_t\}$ until time t and being in state i at time t, given the model λ. α is calculated as follows:

$$\alpha_1(i) = \pi_i b_i(O_1) \tag{3.13}$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{Q} \alpha_t(i)a_{ij}\right] b_j(O_{t+1}) \tag{3.14}$$

Similarly the backward variable $\beta_t(i)$ is defined as the probability of being in state i at time t and observing the partial sequence $\{O_{t+1}...O_T\}$. β is calculated as follows:

$$\beta_T(i) = 1 \tag{3.15}$$

$$\beta_t(i) = \sum_{j=1}^{Q} a_{ij}b_j(O_{t+1})\beta_{t+1}(j) \tag{3.16}$$

After calculating the forward and backward variables, γ, ξ and $P(O|\lambda)$ are cal-

culated using Equations 3.17 and 3.18:

$$\gamma_t(i) = \sum_{j=1}^{Q} \xi_t(i,j) \tag{3.17}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(l)} \tag{3.18}$$

In the M-step, the transition coefficients and the initial states probabilities are re-calculated, whereas the observation probabilities are updated by estimating the means and the covariances of the Gaussians. If there are K observation sequences, then the new parameters are the averages over all observations in all sequences. The new parameters are computed as follows:

$$\pi_i = \frac{\sum_{k=1}^{K} \gamma_1^k(i)}{K} \tag{3.19}$$

$$a_{i,j} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \xi_t^k(i,j)}{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \gamma_t^k(i)} \tag{3.20}$$

$$\mu_i = \frac{\sum_{t=1}^{T} \gamma_t(i) O_t}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.21}$$

$$\Sigma_i = \frac{\sum_{t=1}^{T} \gamma_t(i)(O_t - \mu_i)(O_t - \mu_i)^T}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.22}$$

The iteration continues until a certain iteration number is achieved or the likelihood converges. In order to calculate the likelihood of the training set, the following equation is used:

$$\mathfrak{L} = \sum_{k=1}^{K} \log \left(\sum_{i=1}^{Q} \alpha_T^k(i) \right) \tag{3.23}$$

where K is the number of observation sequences and Q is the number of states.

When the model parameters are found, the state sequence $Q = \{q_1 \dots q_T\}$ of an observation sequence $O = \{O_1 \dots O_T\}$ can be computed using the Viterbi algorithm. Here, $\delta_t(i)$ is defined as the probability of the highest probability path at time t that accounts for the first t observations and ends in state i:

$$\delta_t(i) \equiv \max_{q_1 \dots q_{t-1}} P(q_1 \dots q_{t-1}, q_t = S_i, O_1 \dots O_t | \lambda) \tag{3.24}$$

The optimal path q^* representing the state sequence can be found recursively as in algorithm 3.5.

Initialization

$\delta_1(i) = \pi b_i(O_1)$;

$\psi_1(i) = 0$;

Recursion

$\delta_t(j) = \max_i \delta_{t-1}(i) a_{ij} b_j(O_t)$;

$\psi_t(j) = \arg\max_i \delta_{t-1}(i) a_{ij}$;

Termination

$q_t^* = \arg\max_i \delta_T(i)$;

Backtracking

$q_t^* = \psi_{t+1}(q_{t+1}^*, t = T-1, T-2, \dots, 1$;

Figure 3.5. Viterbi Algorithm.

For recognition, the likelihoods of a sequence for several HMM models are computed using Equation 3.23. The model with the highest likelihood is then selected by the recognizer.

3.2.2. Coupled Hidden Markov Models

Coupled hidden Markov model (cHMM) is a generalization of HMM that integrates two or more streams of data. Here, a collection of HMMs are combined such that the discrete states of each HMM are conditioned by the discrete states of all the related HMMs. As a consequence, cHMMs are able to model data streams that are correlated, as it is the case when data come from different modalities. Figure 3.6 illustrates the dependency graph of a bimodal coupled HMM.

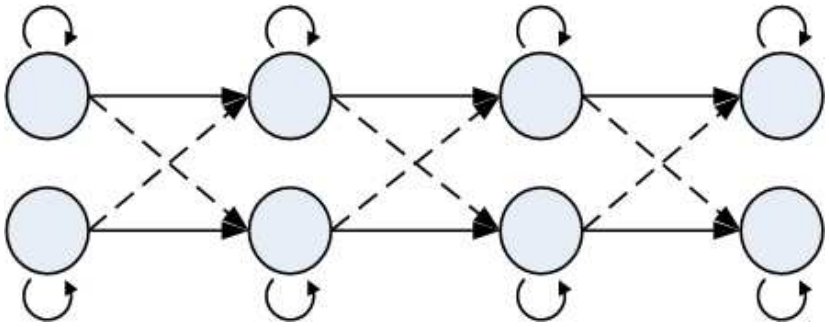


Figure 3.6. Dependency graph of a bimodal coupled HMM.

The coupling algorithm is based on reducing the K-modal coupled HMM to an ordinary but complex HMM by computing the Cartesian product of all sub-HMM parameters. The hidden state transition probabilities of the cHMM can be computed as follows:

$$P(q_t|q_{t-1}) = \prod_{n=0}^{K-1} P(q_t^n|q_{t-1}^0, q_{t-1}^1, ..., q_{t-1}^{K-1}) \tag{3.25}$$

where K is the number of modalities, q_t is the the state variable of the cHMM at time t, and q_t^n is the state variable of the HMM for modality n. In the end, the parameters of the HMMs can be found by projecting back on the HMM space.

Considering that each state depends on the previous states of each modality, if two HMMs having state numbers $N1$ and $N2$ are coupled, then the sub-HMMs

transition matrices will have dimensionality $N1 \times N1 \times N2$ and $N2 \times N1 \times N2$. Here, the transition probability can be given as:

$$q_t^1(i,j,k) \equiv P(q_t^1 = s_i^1 | q_{t-1}^1 = s_j^1, q_{t-1}^2 = s_k^2) \tag{3.26}$$

where q_t^1 is the state variable of the first HMM at time t and s_j^1 is the probability of being in state i of the first HMM at time t. By taking the Cartesian product, the $(N1 \times N2) \times (N1 \times N2)$ state transition matrix of the coupled HMM can be computed. Then, the transition probability q_t of cHMM at time t is given as:

$$q_t(i,j,k,l) \equiv P(q_t^1 = s_i^1, q_t^2 = s_j^2 | q_{t-1}^1 = s_k^1, q_{t-1}^2 = s_l^2) \tag{3.27}$$

A similar Cartesian product argument is used to construct the space of observations. The obtained cHMM is than considered as a continuous HMM having $(N1 \times N2)$ states, and usual computations for continuous HMM follows in the E-step. At the M-step, cHMM parameters are projected back and the sub-HMM parameters are updated. The back-projection is established as follows:

$$\begin{aligned} P(q_t^1 = s_i^1 | q_{t-1}^1 = s_k^1, q_{t-1}^2 = s_l^2) &= \sum_{j=1}^{N2} P(q_t^1 = s_i^1, q_t^2 = s_j^2 | q_{t-1}^1 = s_k^1, q_{t-1}^2 = s_l^2) \\ &= \sum_{j=1}^{N2} P(q_t^1 = s_i^1 | q_{t-1}^1 = s_k^1, q_{t-1}^2 = s_l^2) \\ &\quad P(q_t^2 = s_l^2 | q_{t-1}^1 = s_k^1, q_{t-1}^2 = s_l^2) \\ &= P(q_t^1 = s_i^1 | q_{t-1}^1 = s_k^1, q_{t-1}^2 = s_l^2) \\ &\quad \sum_{j=1}^{N2} P(q_t^2 = s_l^2 | q_{t-1}^1 = s_k^1, q_{t-1}^2 = s_l^2) \end{aligned} \tag{3.28}$$

$$\tag{3.29}$$

In our study we model each sequence with a left-to-right HMM model and a left-to-right coupled HMM model where the state number is proportional to the number of frames in the sequence. In coupled HMMs the motion and shape is modeled as different

processes and coupled to model the loose synchronization in between. In each case, the training is performed by leave-one-out cross validation, with one example in the test set and remaining examples in the training set. We assume that in each sequence, there are frames at the start and end of the sequence that are junk frames and unrelated to the sign. So, in our model each word has an HMM model containing two junk states, one in the beginning and one in the end. We train an HMM for each word, such that the starting state and the end state is common for all the sequences of the word. This is shown in Figure 3.7.

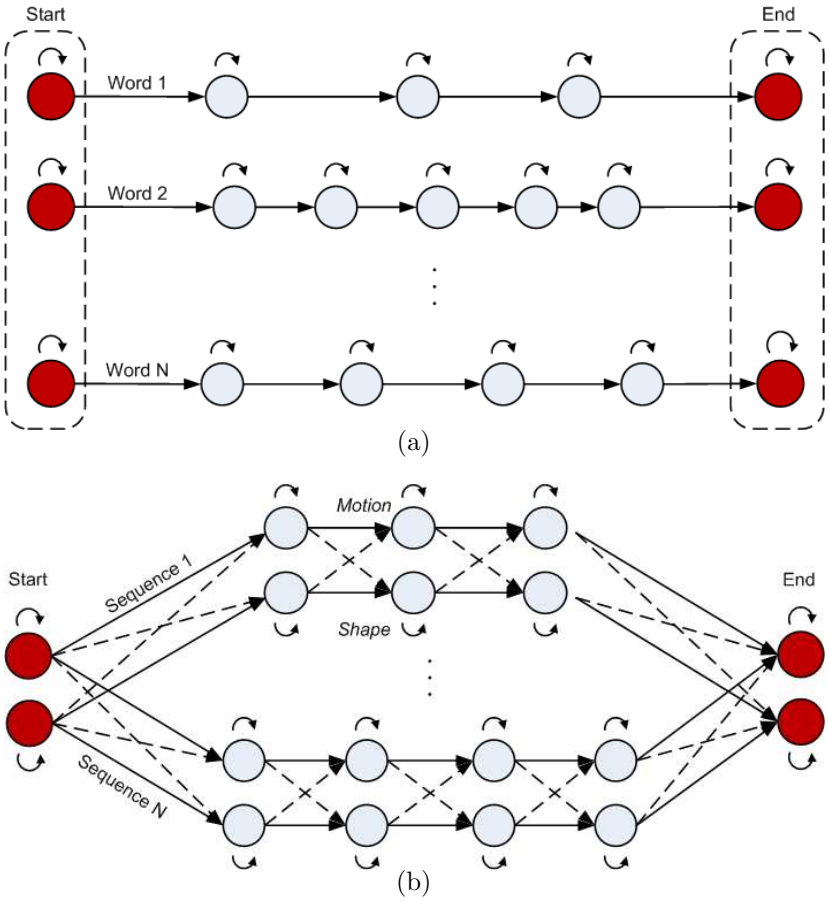


Figure 3.7. Dependency graphs of HMM. (a) The dependency graph of left-to-right HMM, and (b) the dependency graph of coupled HMM.

After modeling the HMMs we aim to find the locations when the sign starts and ends. There are two approaches to obtain this information. In the first approach we compute the state sequence for each sign and find the locations when the observations

come from the junk states. Then we take the frame where the first junk state ends as the start frame and the frame where the last junk state starts as the end frame.

In the second approach we compute the probabilities of being in the first and last state for each observation of a given sequence using the Equation 3.18. Then, by thresholding these values we find the frames when the junk states end.

3.3. Fusion Techniques

In our study, we are dealing with hand gestures in sign language where we concentrate on two modalities, hand motion and hand shape. Both of these modalities carry important information about the gesture and combining them according to their synchronization with each other can improve the performance.

As a first attempt, we concatenate the feature vectors of all the concurrent modalities into a single feature vector. This method assumes that the modalities are in full synchronization. In the end we obtain 26 feature sets by combining the shape and motion features. The obtained feature sets can be seen in Table 3.1.

Table 3.1. Feature sets obtained by feature level fusion.

Initial Feature sets	Combination with CoM	Combination with CoM, △CoM	Combination with CoM, △CoM, E
CoM			
△CoM	CoM, △CoM		
E	CoM, E	CoM, △CoM, E	
RDF	CoM, RDF	CoM, △CoM, RDF	CoM, △CoM, E, RDF
H	CoM, H	CoM, △CoM, H	CoM, △CoM, E, H
DCT	CoM, DCT	CoM, △CoM, DCT	CoM, △CoM, E, DCT
HOG	CoM, HOG	CoM, △CoM, HOG	CoM, △CoM, E, HOG
LBP	CoM, LBP	CoM, △CoM, LBP	CoM, △CoM, E, LBP

In the second approach, each modality is modeled by a different processes and in the end, their results are fused. As discussed in the previous section, in the coupled HMM approach, each modality is coupled with established links between the states

of different processes. We also tested the performance of parallel HMMs where two independent and parallel HMMs are trained for each sign. In the end we obtain different segmentation results from the sub-HMMs. These results are then combined such that the frame where the first state ends in both of the processes is taken as the start frame and the frame where the last state starts is taken as the end frame. To evaluate the performance, shape and motion features are used to train cHMM and parallel HMM as in Table 3.2.

Table 3.2. Feature sets used in cHMM and parallel HMM.

HMM 1	HMM 2
CoM	E
CoM	RDF
CoM	H
CoM	DCT
CoM	HOG
CoM	LBP
CoM, △CoM	E
CoM, △CoM	RDF
CoM, △CoM	H
CoM, △CoM	DCT
CoM, △CoM	HOG
CoM, △CoM	LBP
CoM, △CoM, E	RDF
CoM, △CoM, E	H
CoM, △CoM, E	DCT
CoM, △CoM, E	HOG
CoM, △CoM, E	LBP

In the last method, we combine HMM and DTW where we train HMMs on the intervals that are found by the DTW algorithm. When the interval including the sign is wide, a high number of observations correspond to junk states while the inner states are expected to represent relatively short sequences. In these cases, most of the observations that do not belong to the sign are included in the segmented part which decreases the performance. In our experiments we have seen that DTW can serve for narrowing the search window. Therefore, we first apply DTW to the sequences to reduce the search interval and then apply HMM to find the segmentation.

4. EXPERIMENTAL RESULTS

In this chapter, we provide the results of our alignment experiments performed on a database of Turkish signed speech videos. In Section 4.1, we introduce the database and discuss its challenges. In Section 4.2, we compare our alignment approaches and fusion techniques for each of our feature extraction methods. We also analyze the results by taking the challenges of our database into account.

In order to measure our segmentation accuracy, in Section 4.3, we examine the effect of the segmentation results in recognition of the signs. We further discuss the effect of our feature extraction methods on the recognition task.

4.1. Database

In this study, we use a database of Turkish signed speech videos of TRT broadcast news for the hearing impaired. The database contains 15 videos. In all of the videos, the same newscaster is presenting the news by speaking and signing simultaneously. The total length of the videos is around two hours, with 174,939 frames and a total of 10,318 words. These words correspond to 3,498 different signs. The exact start and end locations of the signs are manually annotated by TSL signers. The spoken term detection accuracy on these videos is shown to be 85.7 per cent. To measure the performance of the tracking, the ground truth for the center of mass coordinates of the hand and face is manually annotated for 15 minutes of these videos. The tracking performance on this ground truth data is 99 per cent for the face and 96 per cent for the two hands.

In our experiments, we use a subset of this database including 1,200 sign samples in total. To obtain this set, we selected 40 words, among the most frequent ones, from the whole database, such that for each word there are 30 sign samples. For 20 of the selected words, the corresponding signs are one handed and for the other 20, the signs are two handed. Since the presenter is right handed, the one-handed signs are

performed with the right hand. The detailed information for the selected signs can be seen in Table 4.1.

The words in this set are successfully segmented by the spoken term detection with a performance of 100 per cent. According to this information, the average duration of the corresponding speech interval is 15.99 frames. In Table 4.1, we show the average duration for each sign with respect to manual annotation. Considering the ground truth intervals, the average sign duration is found to be 15.72 frames.

To measure the tracking performance, we considered each sign sample and denoted the tracking accuracy of a sample as erroneous, when the tracking error continues for more than three frames. Consequently, the tracking accuracy on these signs is found to be 98.58 per cent for the two hands, and 100 per cent for the face. Number of erroneous samples for each word is given in Table 4.1.

We further analyzed the signs with respect to the occlusions and contact of the hands with each other or with the face. When one hand is located in front of the face such that the hand image is fully included in the face area, we denote the situation as full head occlusion. Similarly, when one hand cannot be seen due to the other hand, we refer to the situation as full hand occlusion. On the other hand, when only a contact is observed between the two hands or the hand and the face, we call it partial occlusion. Examples for occlusions can be seen in Figure 4.1.

For our performance analysis, we classified the sign samples according to these occlusion types. When we observe that one type of occlusion takes place for more than three frames, we classified the sign sample accordingly. We observed that with images containing partial occlusion, our segmentation approach result in noisy outer contours. Moreover, in cases where full occlusion is detected, the hand cannot be separated from the background and segmentation result contains both the hand and the object that is in occlusion with the hand. In the selected database, 13.75 per cent of the samples contain partial head occlusion, and five per cent of the samples contain full head occlusion. Partial hand occlusion occurs in 29.67 per cent of the samples,



Figure 4.1. Occlusion examples in the database. (a) partial occlusion of head and hand, (b) full occlusion of head and hand, (c) partial occlusion of hands, (d) full occlusion of hands.

whereas in 5.58 per cent of the samples, the hand is in full occlusion with the other hand. We give the number of occluded samples for each word in Table 4.1. Totally, 47.83 per cent of the samples contain at least one type of occlusion. Having nearly half of the signs with occlusion or contact, we can describe our database as a challenging one as dealing with occlusions and contacts is difficult both in tracking, feature extraction and alignment steps.

4.2. Alignment

The performance of our segmentation depends on several parameters, such as the duration of the samples, the size of the intervals containing the sign, occlusion, the number of the hands that are involved in signing, and features that are used. In this section, we will analyze the effects of these parameters and compare our approaches.

We use four different performance measures to evaluate the system performance: accuracy, precision, recall, and overlapping rates. To calculate these measures we compare the interval of the sign found by the algorithm with the ground truth interval via the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative

Table 4.1. Database.

			Head Occlusion		Hand Occlusion		Tracking Error	
	Words	Duration	partial	full	partial	full	Left Hand	Right Hand
One-handed signs	yol	10.13	0	0	16	1	0	0
	Irak	13.87	0	30	3	0	0	4
	ölmek	14.10	0	0	18	6	0	0
	aday	18.03	1	0	16	0	0	0
	almak	13.20	3	0	1	0	0	0
	anayasa	19.83	0	0	10	0	2	0
	Ankara	15.83	30	0	0	0	0	0
	başbakan	18.03	0	0	30	0	0	0
	bulmak	13.97	0	0	2	0	0	0
	gün	10.80	7	0	0	0	0	0
	gelmek	12.90	0	0	9	0	0	0
	yıl	11.60	30	0	0	0	0	0
	yapmak	13.90	0	0	0	0	0	0
	söylemek	19.67	30	0	8	21	0	0
	Türkiye	18.83	0	30	0	0	0	4
	cumhurbaşkanı	30.03	30	0	11	0	0	0
	ve	10.83	0	0	0	0	3	0
	gerekmek	13.57	0	0	0	0	0	0
	genel	20.60	0	0	0	0	0	0
	istemek	13.87	0	0	0	0	0	0
Two-handed signs	karşı	14.47	0	0	0	0	1	0
	meclis	17.60	0	0	30	0	0	0
	seçim	19.60	0	0	0	0	0	0
	çıkmak	12.77	0	0	0	30	0	1
	ülke	13.50	0	0	1	0	0	0
	açıklamak	13.67	0	0	30	0	0	0
	kişi	12.13	0	0	30	0	0	0
	ara	14.93	0	0	0	0	0	0
	birlik	14.53	0	0	30	0	1	1
	düzen	14.40	0	0	0	0	0	0
	değişmek	18.60	0	0	30	0	0	0
	halk	12.50	0	0	21	9	0	0
	için	13.03	0	0	30	0	0	0
	karar	17.87	30	0	30	0	1	1
	vermek	13.40	0	0	0	0	0	0
	terör	17.50	0	0	0	0	0	0
	toplantı	22.20	0	0	0	0	0	0
	olmak	16.57	0	0	0	0	0	0
	görüşmek	19.10	4	0	0	0	0	0
	sonra	16.87	0	0	0	0	0	0
	TOTAL	15.72	165	60	356	67	8	11

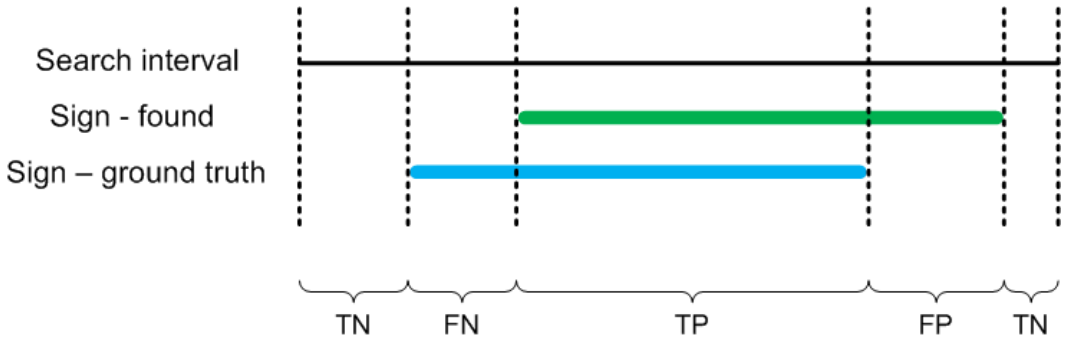


Figure 4.2. The True Positive (TP), True Negative (TN), False postive (FP) and False Negative (FN) values for the extracted sign with respect to the search interval and the ground truth.

(FN) values. TP value is the length of the interval that is found by intersecting the ground truth interval and the found interval. Hence, TP represents the interval which is correctly found by the algorithm to include the sign. Similarly, TN value stands for the sum of the intervals that are located at the beginning and end of the search interval and correctly excluded by the algorithm for not including the sign. FP and FN values are the errors made by the algorithm. In Figure 4.2, an illustration of these values can be seen. Equations 4.1 - 4.4 show the calculation of the performance measures.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

$$\text{Overlapping} = \frac{TP}{TP + FN + FP} \tag{4.4}$$

Using the performance measures, we compute the performances for each sign and report their mean over the whole database. Moreover, we group the signs in the database according to the information on occlusion, duration, and number of hands involved in signing, and compare the results. To compare the performances on the occluded images, we divided the database into two, where the signs having more than 15 samples with occlusion are grouped in occluded sign data set and the rest in the non-

occluded sign data set. The signs having a duration less than 15 frames are denoted as short signs. We compare our methods according to these groups.

Since the speech and signing is not fully synchronized, it is possible that the intervals coming from the speech recognizer do not include all of the signing. To guarantee that the intervals include the sign, we enlarge the intervals from the start and the end, and use the enlarged intervals for alignment. In our database, signing starts approximately 15 frames earlier than the spoken word. When we enlarge our intervals by 21 frames from the start and six frames from the end, all samples are included wholly in the enlarged intervals. On the other hand, when we enlarge our intervals by 11 frames from the start and shorten by five frames from the end, i.e. when we use the parameters $(+11, -5)$ for enlarging the intervals, 39.75 per cent of the signs are larger than the intervals. The percentages of the samples for several intervals are shown in Table 4.2.

Table 4.2. Percentages of the samples with respect to different enlargement parameters, and performance of HMM using the feature set CoM, $\triangle$CoM, E.

Enlargement Parameters		Included Samples				
start	end	(%)	Acc	Pre	Rec	Ovr
21	6	100.00	65.79	55.90	93.47	53.07
20	5	99.50	65.82	56.78	92.86	53.70
19	4	98.92	67.10	58.73	93.73	56.05
18	3	98.42	68.26	60.97	93.71	57.89
17	2	97.25	69.75	63.50	93.85	60.35
16	1	96.17	71.01	65.91	93.31	62.18
15	-1	93.00	72.75	69.66	93.12	65.31
14	-2	88.08	74.21	72.42	92.24	67.51
13	-3	81.33	75.73	75.45	91.19	69.67
12	-4	72.50	77.31	78.41	90.74	72.04
11	-5	60.25	78.23	81.13	89.42	73.54

As the interval gets narrower, FP value decreases as a result of the shortening of the intervals that do not contain signs. Consequently, the precision value increases. Moreover, the performance of the HMM improves when the size of the observations

coming from the junk state decreases. Therefore, when the search area is shortened, the accuracy values increase. In our study we experiment with the intervals that are enlarged by the parameters: $(+21, +6)$ and $(+15, -1)$.

As discussed in Chapter 3, we use two methods to find the start and end locations for the HMM method. In the first approach, we find the Viterbi path and choose the first observation after the first junk state as the starting location, and the last observation before the second junk state as the end point. As a second approach, we compute the probabilities of being in the junk states, and by thresholding these values, we decide when the junk states are left or entered. We observed that using the first approach, we obtain better performance. In Figure 4.3, we plot the precision-recall curve obtained with the threshold values between 0.01 and one for the feature CoM, $\triangle$CoM, E using the enlargement parameters $(+15, -1)$. .

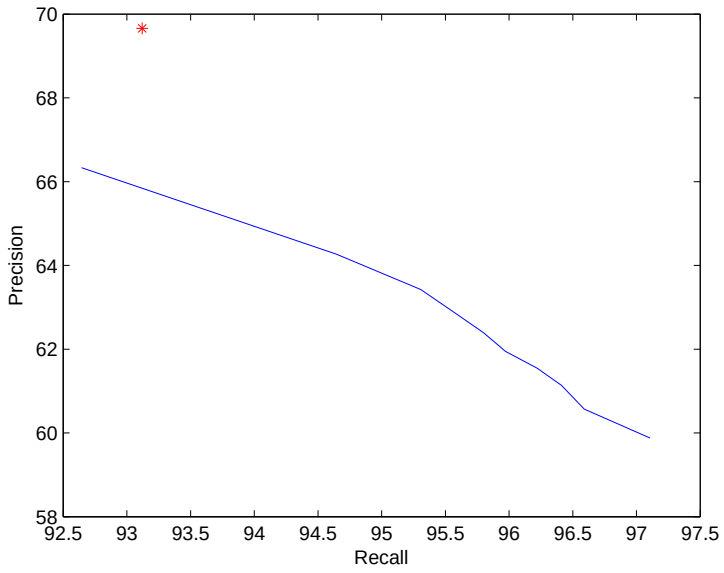


Figure 4.3. Precision-recall curve for the feature CoM, $\triangle$CoM, E. The red point signifies the result found using the Viterbi path.

Exact match with the ground truth is rarely possible due to the uncertainty of the sign boundaries. Consequently, a lower bound for the the accuracy is needed

to evaluate the performances. Therefore, as a lower bound, we compute the accuracy values for the cases, when the whole interval that is found by the spoken term detection is used as the found interval without any further computations. Here, TP values will reflect the lengths of the exact signs, whereas TN values will be zero. According to this computation, when we enlarge the interval by 15 frames in the beginning and shorten by one frame at the end, the average accuracy value for our database becomes 51.73 percent. Moreover, when we enlarge the interval by 21 frames at the beginning and six frames at the end, the average accuracy becomes 36.29 per cent.

As discussed in Chapter 2, we concatenate the feature vectors belonging to the right hand and the left hand. The main reason for this approach is the fact that clustering the signs according to the number of involved hands is a challenging task. When the information of the left hand does not change the meaning of the sign, the sign is denoted as one-handed. However, in most of the one-handed signs, the left hand remains stationary. Therefore, we observed that using the information of both hands does not change the accuracy distinctively, even when the sign is one-handed.

In Table 4.3, we compare the accuracies of using only the right hand and the two hands for our feature extraction methods. For alignment we use HMM in the search interval enlarged by 15 frames in the beginning and reduced by one frame at the and. We compare the accuracy means over the whole set, one-handed sign set, and two-handed sign set. It can be seen that the performances obtained on the whole set are very close. Obviously, using only one hand improves the performance when all of the signs in the database are one-handed, whereas using both of the hands gives a better performance in a database including only two-handed signs. However, considering all the signs, using information of both of the hands performs slightly better. Therefore, in our experiments we use the information of both of the hands.

In our experiments, we first compare the performances of DTW and HMM for different feature methods using paired t-test. We also examine the results of feature level fusion. Recall that we use features to represent motion: CoM, $\triangle$CoM; geometric shape features: E, RDF, H; and shape features that use texture information: DCT,

Table 4.3. Performance of using only the right hand and the two hands.

	one hand (%)			two hands (%)		
Feature sets	all signs	one-handed signs	two-handed signs	all signs	one-handed signs	two-handed signs
CoM	68.93	69.37	68.50	71.47	70.76	72.18
△CoM	72.88	72.93	72.82	70.81	69.11	72.50
E	69.55	69.78	69.33	69.36	68.27	70.46
RDF	68.91	69.54	68.29	65.13	64.20	66.07
H	59.17	60.61	57.73	59.89	60.13	59.64
DCT	56.54	56.53	56.54	56.98	57.36	56.60
HOG	56.56	56.65	56.47	57.21	57.26	57.16
LBP	55.24	55.85	54.64	55.27	55.68	54.86

HOG, LBP. We further apply feature level fusion to combine the motion and the shape information. The results of our experiments for DTW and HMM can be seen in Tables 4.4 and 4.5. In Table 4.4, the results obtained with the enlargement parameters $(+15, -1)$ are shown, whereas in Table 4.5, the results using the enlargement parameters $(+21, +6)$ are given.

In Tables 4.4 and 4.5, we see that when we do not combine the feature vectors, using motion information alone gives the best performance, whereas ellipse information and speed information follow with small decrease in the performance. When we compare shape features, we see that the effect of the high level shape descriptors, such as DCT, HOG and LBP is limited. We think that this is due to the low resolution of the hand shapes and also due to the occlusions with the face and the other hand. Geometric shape descriptors such as ellipse features are more robust to low resolution and occlusions.

When we apply feature level fusion, in DTW, we observe a slight change in the performances for different feature extraction methods. We see that in DTW, not combining the features and using CoM, △CoM features alone performs consistently better. Moreover, when the search window is larger, even if the performance is better, recall values increase, whereas precision values and overlapping ratios decrease. From this result we can deduce that for larger search intervals DTW algorithm returns larger

intervals and consequently false positive values increase while false negative values decrease. Since the overlapping ratio decreases, we can conclude that for larger intervals DTW algorithm is more erroneous and the increase in the performance is due to the increase in TN values.

Table 4.4. Performance of DTW and HMM with respect to accuracy, precision, recall and overlap in the enlarged window with $(+15, -1)$.

	DTW (%)				HMM (%)			
Feature sets	Acc	Pre	Rec	Ovr	Acc	Pre	Rec	Ovr
CoM	77.99 ± 2.51	77.55	86.75	68.23	71.47 ± 1.72	68.33	93.73	64.45
△CoM	75.31 ± 1.83	83.85	70.45	61.20	70.81 ± 2.25	67.64	91.65	63.56
E	77.98 ± 2.04	78.09	85.21	68.11	69.36 ± 2.50	68.36	82.78	59.33
RDF	75.16 ± 2.23	75.56	85.83	65.72	65.13 ± 2.18	62.59	93.44	59.25
H	76.61 ± 2.13	78.73	80.16	65.56	59.89 ± 2.37	60.54	76.43	47.85
DCT	75.37 ± 2.46	73.52	87.15	66.01	56.98 ± 2.51	57.14	76.16	45.02
HOG	75.79 ± 2.33	74.68	86.01	66.15	57.21 ± 2.00	58.26	75.65	44.95
LBP	75.47 ± 2.24	74.17	86.26	65.88	55.27 ± 2.36	62.25	56.80	35.52
CoM, △CoM	$\mathbf{79.69 \pm 2.20}$	85.01	79.37	68.08	71.20 ± 1.82	67.80	94.71	64.69
CoM, E	79.07 ± 2.07	78.84	86.78	69.32	72.82 ± 2.37	70.88	90.09	64.74
CoM, RDF	77.12 ± 2.32	78.15	85.29	67.30	68.76 ± 1.89	65.13	95.64	62.63
CoM, H	78.52 ± 2.40	78.41	86.15	68.74	70.81 ± 2.00	70.12	88.55	62.03
CoM, DCT	78.28 ± 2.31	76.45	88.81	69.04	59.10 ± 1.66	63.38	79.05	47.67
CoM, HOG	78.04 ± 2.37	76.54	88.36	68.73	58.62 ± 1.52	62.74	78.39	47.05
CoM, LBP	78.12 ± 2.25	76.66	88.22	68.75	57.40 ± 2.36	69.15	61.50	38.93
CoM, △CoM, E	79.51 ± 1.61	80.18	85.79	69.49	72.75 ± 2.11	69.66	93.12	65.31
CoM, △CoM, RDF	77.05 ± 2.14	78.29	85.16	67.15	68.00 ± 1.76	64.37	95.86	62.10
CoM, △CoM, H	79.02 ± 1.99	81.02	83.58	68.56	70.97 ± 2.51	69.42	90.52	62.87
CoM, △CoM, DCT	79.25 ± 2.01	79.47	86.33	69.53	58.85 ± 1.98	62.29	80.88	48.30
CoM, △CoM, HOG	78.83 ± 2.30	79.01	86.24	69.03	58.74 ± 1.93	62.77	79.16	47.49
CoM, △CoM, LBP	78.88 ± 2.07	78.94	86.28	69.09	57.28 ± 2.86	68.74	60.73	38.46
CoM, △CoM, E, RDF	77.90 ± 2.18	78.60	86.05	68.16	69.63 ± 1.91	66.19	95.28	63.27
CoM, △CoM, E, H	78.74 ± 1.90	79.21	85.70	68.83	$\mathbf{73.16 \pm 1.95}$	71.71	90.31	64.86
CoM, △CoM, E, DCT	78.81 ± 1.99	78.51	86.93	69.25	59.82 ± 1.89	63.57	81.50	49.31
CoM, △CoM, E, HOG	78.61 ± 1.97	78.01	87.47	69.10	59.79 ± 1.81	63.30	80.56	48.88
CoM, △CoM, E, LBP	78.32 ± 1.92	77.91	86.87	68.70	57.84 ± 2.24	68.85	64.64	40.48

In HMM, combining motion features with simple shape descriptors such as ellipse parameters, RDF or Hu moments improves the performance. We see a distinctive decrease in the performance when high level features, such as DCT, HOG and LBP

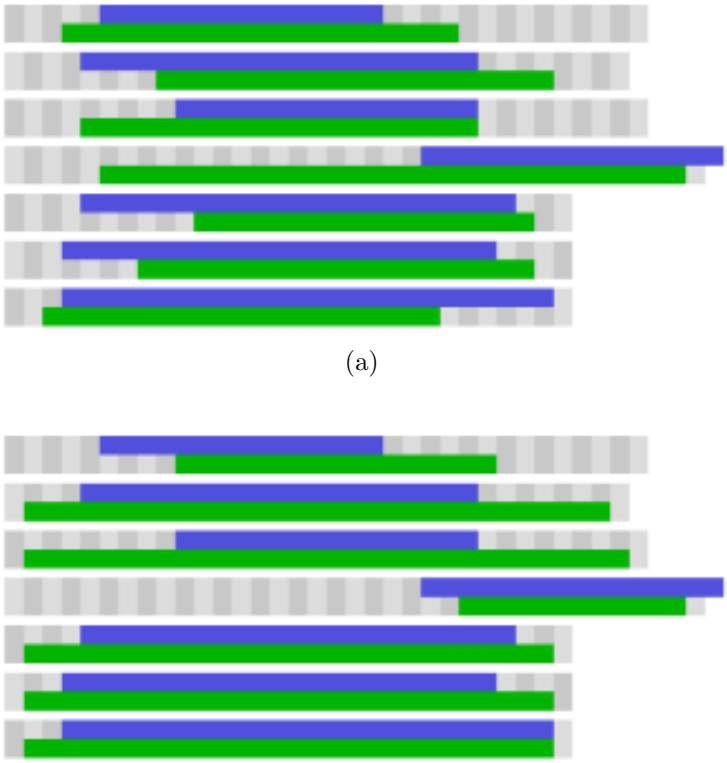
are used. For larger search intervals we obtain a decrease in accuracy, precision and overlapping ratio, whereas the change in recall rates is very small. The reason for this is the increase in false positive values. The high recall rates show that the intervals found by the HMM algorithm are always larger than the original interval of the sign. Since the performances obtained using high level features are close to the lower bounds, we can say that using HMM, we obtain meaningful results only when we use simple shape features and motion.

We see that the performance of DTW is significantly better with respect to HMM with 95 per cent confidence. However, when the synchronization between the speech and the sign is poor, HMM detects the beginning and end points of the sign more accurately. An example for this case can be seen in Figure 4.4, where we illustrate six alignment results of DTW and HMM for the word "prime minister".

For fusing the shape features with motion parameters, we apply two methods: coupled HMM and parallel HMM. In coupled HMM, we combine two HMMs, one for modeling the motion and the other for modeling the shape, at the training level in such a way that their discrete states are conditioned by the states of both of the HMMs. Consequently, the correlation between these two modalities are also modeled. In parallel HMM, we train independent HMMs for each modality and combine their result as discussed in Chapter 3. The results of our experiments for coupled HMM and parallel HMM for different enlargement windows can be seen in Tables 4.6 and 4.7.

In Table 4.6 we see that the performances obtained by coupled HMM and parallel HMM are very close, however in most of the cases with coupled HMM a slight improvement is observed. The best performance is obtained using coupled HMM with an accuracy of 75.11 per cent by combining CoM, $\triangle$CoM features with ellipse parameters. We can state that, with 95 per cent confidence, coupled HMM performs significantly better than parallel HMM.

When we compare these results with left-to-right HMMs, we see that a significantly better performance is obtained, especially for high level features. There is an



(a)

(b)

Figure 4.4. Six examples for the alignment of the word “prime minister” for (a) DTW (b) HMM, using the feature CoM, △CoM, E. The sign is searched within the gray area. Each box stands for one frame, the green lines represent the found segment, and the blue lines represent the ground truth.

increase in precision values and decrease in recall values, which can be explained by a decrease in false positives and an increase in false negatives. This results in a slight decrease in the overlapping ratios. However, the accuracies are always higher than 60 per cent and this shows that the segmentation is much better compared to left-to-right HMMs.

In Table 4.7, we see that enlarging the intervals changes the accuracy performances slightly, whereas a marked decrease in the overlapping ratios and the precision values are observed. Hence we can deduce that for larger search windows, larger intervals are returned. Since both the false positives and the true positives increase, the accuracy values are nearly the same.

As already discussed, when the synchronization between speech and sign is poor, HMM segments the sign better compared to DTW. Therefore, we further combine these algorithms using sequential fusion to increase the performance. Here we first find intervals using DTW and then apply HMM (DTW-HMM), parallel HMM (DTW-pHMM)or coupled HMM (DTW-cHMM) on these intervals to correct the instances with poor synchronization. The results of HMM after DTW are shown in Table 4.8.

Here we observe an increase in the accuracies compared to DTW. There is an increase in the precision values, and decrease in recall values. This is surprising considering the high recall values in HMM. It is a pleasing result, since the difference between the recall and precision values decrease. When the search interval is narrow, a small decrease in the overlapping ratios is observed in the case of DTW-HMM method. However, this is not the case for larger search intervals. Considering the accuracy values, DTW-HMM performs significantly better than DTW with 95 per cent confidence. Hence we can infer that DTW-HMM approach gives better segmentation results.

We also combine DTW with coupled HMM and parallel HMM. Surprisingly DTW-HMM method performs generally better than DTW-cHMM and DTW-pHMM, as shown in Tables 4.9 and 4.10. For narrow search window, the accuracies are even worse than the results obtained by DTW with the exception of the RDF feature. The

best accuracy among all the methods is 81.24 per cent, obtained with DTW-pHMM method by combining the CoM, $\triangle$CoM, E feature with RDF. Yet, for larger search intervals, the best accuracy among all the methods is 84.61 per cent, obtained with DTW-HMM method using the CoM, $\triangle$CoM, E feature.

When we analyze the results in Table 4.10, we see that applying coupled HMM or parallel HMM after DTW increases the accuracy compared to DTW only when the search window is large. However, overlapping ratio is generally smaller, which is not the case for the DTW-HMM method. Therefore, we believe that the increase in the accuracies is misleading and we cannot conclude that the segmentation improves with the use of coupled or parallel HMMs.

Considering these results, we can conclude that for segmentation the best approach is to combine DTW and HMM. To explain the hand gesture, the motion and speed features are necessary, whereas for describing the shape, ellipse parameters give the best performance.

We further analyze the effect of occlusion, duration and number of hands involved in signing. Since we obtain similar results for the number of hands and occlusion in all the used methods, we will only report the results for DTW-HMM method for the enlarged window $(+21, +6)$. The result can be seen in Table 4.11.

Considering the number of the hands we see that there is a slight increase in the performances when the sign is two-handed. This was expected, since the two-handed signs contain more information. We know that when the sign is one-handed, the left hand usually remains stationary. Therefore the accuracies of one-handed signs are not very different compared to two-handed signs.

It is an interesting observation that the accuracies are not effected by the occlusion. Moreover, when simple shape descriptors are used, the performance on the occluded signs are better than the non-occluded sign. Hence we can conclude that ellipse parameters and RDF features are robust to occlusion.

In Table 4.12 we compare the accuracies of DTW and DTW-HMM for the enlarged window $(+21, +6)$ according to the duration information. Here we see that DTW generally performs better for long signs. When we apply HMM after DTW, the accuracies on short sequences consistently increase. On the other hand for large sequences, when high level features are used the performance usually decreases, whereas with ellipse, Hu moments or RDF features a slight improvement on the accuracies are observed. This result is observed for all the HMM variants.

4.3. Recognition

In order to analyze the effects of the features we also compared the recognition accuracies. Here, we use left-to-right continuous HMM to model each word, where the training is performed by leave-one-out cross validation. In Table 4.13 we see the results obtained by recognition using the search window enlarged by $(+21, +6)$, ground truth intervals, intervals found by DTW algorithm and intervals found by the DTW-HMM algorithm. Here we see that high level features show better performance for the recognition task and segmentation improves the performance.

The performance obtained by RDF feature is surprisingly always better either combined with motion and speed or not. The best recognition performance is shown to be 98.67 per cent which is achieved by DTW-HMM algorithm using CoM, $\triangle$CoM, RDF feature. Another interesting observation is the low performance obtained using LBP features. The reason for this might be the low resolution and blur in the hand images.

The most important information is that eliminating the junk frames and using the intervals that only contain the sign improves the recognition performance significantly. Moreover, when we compare automatic segmentation with manual annotation, we do not observe a significant difference. Therefore we can claim that our segmentation is accurate.

Table 4.5. Performance of DTW and HMM with respect to accuracy, precision, recall and overlap in the enlarged window with $(+21, +6)$.

	DTW (%)				HMM (%)			
Feature sets	Acc	Pre	Rec	Ovr	Acc	Pre	Rec	Ovr
CoM	78.40 ± 1.63	68.06	89.02	61.65	63.77 ± 2.58	54.06	94.62	51.84
△CoM	78.60 ± 1.69	74.50	72.53	57.33	61.04 ± 1.70	50.95	95.72	49.69
E	78.60 ± 2.03	67.71	91.00	62.54	63.06 ± 2.53	52.28	79.31	45.75
RDF	73.41 ± 2.52	63.90	91.14	58.14	54.24 ± 2.89	45.97	93.20	44.18
H	78.81 ± 1.78	69.43	85.10	61.27	50.07 ± 2.25	39.57	73.74	32.24
DCT	72.73 ± 1.88	59.90	93.65	57.17	47.29 ± 1.63	37.69	75.05	30.98
HOG	73.56 ± 2.03	60.98	93.04	57.81	47.83 ± 2.24	38.84	74.69	31.05
LBP	73.03 ± 1.68	60.51	93.41	57.51	53.23 ± 2.00	44.76	53.19	24.47
CoM, △CoM	$\mathbf{82.40 \pm 1.49}$	76.84	84.14	65.05	62.38 ± 2.81	52.60	94.60	50.84
CoM, E	79.45 ± 1.62	69.09	91.32	63.54	$\mathbf{67.29 \pm 3.00}$	57.49	90.35	53.46
CoM, RDF	76.68 ± 2.29	67.87	90.09	60.97	57.67 ± 3.01	49.15	95.82	47.51
CoM, H	78.80 ± 1.56	68.24	90.58	62.53	61.72 ± 3.56	53.91	83.38	45.73
CoM, DCT	76.49 ± 1.60	64.34	93.86	60.95	49.18 ± 2.75	45.83	77.55	33.55
CoM, HOG	76.50 ± 1.78	64.61	93.84	61.00	49.14 ± 3.04	46.29	76.52	32.92
CoM, LBP	76.33 ± 1.57	64.01	94.22	60.83	53.60 ± 2.33	52.17	57.58	26.96
CoM, △CoM, E	79.91 ± 1.44	70.46	90.59	63.87	65.79 ± 2.02	55.90	93.47	53.07
CoM, △CoM, RDF	76.24 ± 2.25	67.73	89.99	60.46	56.41 ± 2.28	48.31	96.58	46.89
CoM, △CoM, H	79.79 ± 1.54	71.22	87.92	63.00	60.36 ± 2.79	52.22	89.34	47.22
CoM, △CoM, DCT	78.25 ± 1.77	67.68	91.71	62.45	48.62 ± 2.22	44.13	77.76	33.14
CoM, △CoM, HOG	77.73 ± 2.06	67.19	91.94	61.98	48.69 ± 2.54	45.02	77.10	32.88
CoM, △CoM, LBP	77.56 ± 1.90	66.58	92.27	61.77	53.10 ± 2.07	50.77	58.54	27.02
CoM, △CoM, E, RDF	77.44 ± 2.22	68.27	91.04	61.90	58.87 ± 2.30	49.99	95.29	48.09
CoM, △CoM, E, H	78.83 ± 1.56	68.93	90.61	62.78	64.79 ± 2.56	55.92	89.02	50.85
CoM, △CoM, E, DCT	77.88 ± 1.54	67.00	92.40	62.21	49.37 ± 2.29	46.57	78.17	33.86
CoM, △CoM, E, HOG	77.61 ± 1.89	66.71	92.36	61.89	49.72 ± 2.86	47.70	77.68	33.82
CoM, △CoM, E, LBP	77.20 ± 1.77	66.07	92.67	61.49	53.45 ± 2.69	51.69	59.28	27.61

Table 4.6. Performance of coupled and parallel HMMs with respect to accuracy, precision, recall and overlap in the enlarged window with $(+15, -1)$.

Feature sets		Coupled HMM (%)				Parallel HMM (%)			
HMM 1	HMM 2	Acc	Pre	Rec	Ovr	Acc	Pre	Rec	Ovr
CoM	E	74.93 ± 2.41	75.94	76.50	61.31	73.35 ± 2.33	74.08	79.10	61.40
CoM	RDF	74.83 ± 1.94	73.65	84.60	64.17	73.33 ± 2.25	71.75	88.43	64.21
CoM	H	71.52 ± 2.69	71.94	73.18	56.56	67.37 ± 2.68	67.35	72.72	52.33
CoM	DCT	66.95 ± 3.40	64.45	70.23	51.01	66.11 ± 3.94	64.78	72.22	50.63
CoM	HOG	66.64 ± 3.27	64.67	69.07	50.20	66.22 ± 3.91	65.79	71.80	50.57
CoM	LBP	61.54 ± 3.81	61.97	49.30	37.13	61.83 ± 3.88	67.30	53.92	39.27
CoM, $\triangle$CoM	E	$\mathbf{75.11 \pm 2.49}$	75.81	79.35	62.71	73.49 ± 2.15	73.94	79.76	61.77
CoM, $\triangle$CoM	RDF	73.57 ± 1.72	71.42	88.02	64.31	73.24 ± 2.17	71.28	89.58	64.71
CoM, $\triangle$CoM	H	70.84 ± 2.69	71.23	73.19	56.30	67.50 ± 2.64	67.49	73.39	52.64
CoM, $\triangle$CoM	DCT	65.48 ± 3.77	62.69	71.00	49.77	65.88 ± 3.57	64.47	72.75	50.69
CoM, $\triangle$CoM	HOG	65.54 ± 3.74	62.64	70.76	49.58	66.04 ± 3.51	66.04	72.49	50.76
CoM, $\triangle$CoM	LBP	60.65 ± 3.96	60.81	50.76	36.83	61.74 ± 3.76	67.15	54.74	39.66
CoM, $\triangle$CoM, E	RDF	73.84 ± 2.10	71.89	86.92	64.18	$\mathbf{73.99 \pm 2.19}$	72.30	88.36	64.76
CoM, $\triangle$CoM, E	H	71.88 ± 2.09	72.15	74.45	57.59	68.11 ± 2.68	67.88	72.75	52.88
CoM, $\triangle$CoM, E	DCT	67.12 ± 4.16	64.91	70.90	51.07	66.96 ± 3.66	65.49	72.02	51.20
CoM, $\triangle$CoM, E	HOG	66.56 ± 3.99	64.65	70.77	50.53	67.06 ± 3.84	66.74	71.84	51.21
CoM, $\triangle$CoM, E	LBP	62.20 ± 3.68	64.48	51.61	38.57	62.54 ± 3.90	67.79	54.24	40.03

Table 4.7. Performance of coupled and parallel HMMs with respect to accuracy, precision, recall and overlap in the enlarged window with $(+21, +6)$.

Feature sets		Coupled HMM (%)				Parallel HMM (%)			
HMM 1	HMM 2	Acc	Pre	Rec	Ovr	Acc	Pre	Rec	Ovr
CoM	E	$\mathbf{74.17 \pm 2.43}$	64.04	74.20	52.48	$\mathbf{71.10 \pm 2.52}$	60.83	75.79	50.80
CoM	RDF	69.53 ± 2.22	59.03	85.75	52.98	68.66 ± 2.29	58.00	88.55	53.36
CoM	H	67.80 ± 2.60	55.41	71.57	44.74	64.19 ± 2.50	48.87	70.23	40.10
CoM	DCT	63.87 ± 2.50	46.85	68.16	39.07	63.48 ± 2.70	47.95	71.61	40.08
CoM	HOG	63.69 ± 2.65	46.65	66.80	38.35	63.39 ± 2.43	48.89	71.26	39.76
CoM	LBP	63.59 ± 2.88	43.41	48.03	28.74	63.77 ± 2.53	50.24	50.86	30.00
CoM, $\triangle$CoM	E	73.33 ± 2.73	62.60	78.34	53.32	70.48 ± 2.49	59.89	75.50	50.24
CoM, $\triangle$CoM	RDF	67.31 ± 2.83	57.05	87.39	51.72	67.47 ± 2.78	56.57	88.52	52.40
CoM, $\triangle$CoM	H	67.82 ± 2.55	54.74	71.35	44.74	62.92 ± 2.59	47.85	69.72	39.12
CoM, $\triangle$CoM	DCT	61.41 ± 2.08	44.39	70.89	38.01	62.25 ± 2.62	46.58	71.34	39.05
CoM, $\triangle$CoM	HOG	60.98 ± 2.44	43.39	67.85	36.41	62.05 ± 2.49	47.62	70.81	38.75
CoM, $\triangle$CoM	LBP	62.62 ± 1.69	43.68	50.50	28.83	62.90 ± 2.12	49.57	50.70	29.30
CoM, $\triangle$CoM, E	RDF	68.26 ± 2.32	57.61	86.92	52.33	69.35 ± 2.04	58.68	88.02	53.69
CoM, $\triangle$CoM, E	H	68.70 ± 1.93	55.45	70.66	45.24	65.15 ± 2.02	50.04	69.34	40.66
CoM, $\triangle$CoM, E	DCT	63.71 ± 2.15	46.97	68.53	38.82	64.41 ± 2.05	48.92	70.75	40.42
CoM, $\triangle$CoM, E	HOG	63.35 ± 2.45	46.64	67.10	37.74	64.80 ± 2.05	50.01	70.38	40.48
CoM, $\triangle$CoM, E	LBP	63.32 ± 2.45	47.57	49.98	29.25	64.67 ± 1.74	51.36	50.45	30.47

Table 4.8. Performance of DTW-HMM with respect to accuracy, precision, recall and overlap in the enlarged windows with $(+15, -1)$ and $(+21, +6)$.

	DTW-HMM $(+15, -1)$				DTW-HMM $(+21, +6)$			
Feature sets	Acc	Pre	Rec	Ovr	Acc	Pre	Rec	Ovr
CoM	79.32 ± 2.59	85.61	75.96	66.71	82.56 ± 2.09	78.54	78.74	63.86
△CoM	72.37 ± 2.06	87.76	57.85	53.67	78.69 ± 1.87	80.16	60.69	53.01
E	76.59 ± 2.46	86.24	68.35	61.34	82.07 ± 2.26	78.85	74.79	61.49
RDF	75.77 ± 2.20	80.84	76.31	63.48	77.58 ± 2.40	70.40	83.01	59.55
H	72.68 ± 2.26	84.53	61.44	54.75	78.41 ± 2.12	76.35	65.79	53.31
DCT	70.73 ± 4.02	76.66	65.30	53.51	73.20 ± 2.82	64.49	70.81	47.68
HOG	71.11 ± 3.54	79.02	64.29	53.53	74.04 ± 2.60	66.38	69.79	48.30
LBP	65.57 ± 2.87	81.36	49.36	41.87	71.09 ± 2.50	67.38	51.13	36.85
CoM, △CoM	78.29 ± 2.14	90.61	68.86	63.58	84.19 ± 1.47	84.75	75.28	65.09
CoM, E	80.05 ± 2.26	88.33	74.23	66.85	84.25 ± 1.58	81.62	79.48	66.22
CoM, RDF	78.43 ± 2.04	84.77	76.70	66.17	81.36 ± 1.80	76.05	83.65	64.12
CoM, H	78.90 ± 2.57	87.16	73.43	65.47	83.43 ± 1.91	80.38	78.78	64.69
CoM, DCT	75.32 ± 3.77	82.88	71.74	60.20	77.11 ± 2.35	72.38	75.34	53.93
CoM, HOG	75.03 ± 4.04	83.16	70.68	59.51	77.02 ± 2.34	72.68	73.73	53.07
CoM, LBP	69.87 ± 3.66	86.58	56.30	48.36	74.57 ± 2.56	77.28	58.87	43.67
CoM, △CoM, E	$\mathbf{80.45 \pm 2.05}$	88.24	75.42	67.75	$\mathbf{84.61 \pm 1.51}$	81.23	81.78	67.50
CoM, △CoM, RDF	78.41 ± 1.94	84.45	76.94	66.15	80.98 ± 1.99	75.56	83.94	63.59
CoM, △CoM, H	79.01 ± 2.29	88.21	72.46	65.45	83.43 ± 1.70	81.23	77.97	64.79
CoM, △CoM, DCT	75.69 ± 3.32	85.38	69.71	60.16	78.36 ± 2.35	74.83	74.38	55.39
CoM, △CoM, HOG	74.74 ± 4.03	85.51	68.07	58.37	77.57 ± 2.63	75.01	72.62	53.61
CoM, △CoM, LBP	70.25 ± 3.24	87.38	56.26	49.05	74.93 ± 2.29	78.30	58.44	44.11
CoM, △CoM, E, RDF	79.51 ± 1.89	85.62	77.38	67.44	82.37 ± 1.75	77.04	84.33	65.27
CoM, △CoM, E, H	79.32 ± 2.15	87.17	73.89	66.13	83.77 ± 1.46	80.79	80.00	65.84
CoM, △CoM, E, DCT	75.95 ± 2.93	84.72	70.83	60.87	78.83 ± 2.23	75.46	76.10	56.78
CoM, △CoM, E, HOG	75.48 ± 3.34	84.74	70.14	59.99	78.07 ± 2.21	74.83	74.34	54.99
CoM, △CoM, E, LBP	70.48 ± 3.05	86.17	57.91	49.88	75.04 ± 1.72	77.18	60.58	45.23

Table 4.9. Performance of coupled and parallel HMMs after DTW with respect to accuracy, precision, recall and overlap with the enlarged window $(+15, -1)$.

Feature sets		DTW-cHMM $(+15, -1)$				DTW-pHMM $(+15, -1)$			
HMM 1	HMM 2	Acc	Pre	Rec	Ovr	Acc	Pre	Rec	Ovr
CoM	E	77.38 ± 3.05	85.48	67.95	61.32	78.82 ± 2.27	86.31	71.63	64.07
CoM	RDF	78.98 ± 2.42	84.25	75.36	65.75	79.49 ± 2.19	83.37	78.88	67.58
CoM	H	76.03 ± 2.57	84.66	65.98	59.03	76.12 ± 2.72	84.19	67.54	59.54
CoM	DCT	72.75 ± 4.11	79.30	62.92	54.04	73.76 ± 4.26	78.26	66.65	56.30
CoM	HOG	72.30 ± 4.23	79.86	61.76	53.17	74.02 ± 4.22	79.93	66.04	56.38
CoM	LBP	66.39 ± 3.87	76.97	46.36	40.83	67.75 ± 3.38	81.39	50.34	43.82
CoM, $\triangle$CoM	E	78.54 ± 2.14	87.52	69.73	63.54	79.31 ± 1.92	88.72	70.74	64.63
CoM, $\triangle$CoM	RDF	80.33 ± 2.26	85.77	78.07	68.39	80.30 ± 1.97	85.43	78.03	68.36
CoM, $\triangle$CoM	H	77.12 ± 2.46	87.30	66.88	61.04	76.59 ± 2.38	86.90	66.05	59.76
CoM, $\triangle$CoM	DCT	74.26 ± 3.98	81.45	65.36	56.59	74.68 ± 4.12	81.54	66.02	57.33
CoM, $\triangle$CoM	HOG	73.54 ± 3.84	80.81	63.32	55.21	74.96 ± 3.96	82.71	65.40	57.38
CoM, $\triangle$CoM	LBP	66.81 ± 3.99	77.64	47.45	41.93	68.54 ± 3.31	83.89	50.26	44.78
CoM, $\triangle$CoM, E	RDF	$\mathbf{80.81 \pm 2.23}$	84.65	79.40	69.10	$\mathbf{81.24 \pm 2.09}$	86.03	79.01	69.46
CoM, $\triangle$CoM, E	H	77.53 ± 2.15	86.31	67.96	61.66	76.82 ± 2.21	86.13	67.44	60.29
CoM, $\triangle$CoM, E	DCT	74.63 ± 3.72	80.96	65.85	57.09	75.11 ± 4.15	81.22	66.45	57.66
CoM, $\triangle$CoM, E	HOG	73.97 ± 4.38	81.26	63.79	55.83	75.28 ± 4.18	83.40	65.97	57.77
CoM, $\triangle$CoM, E	LBP	67.87 ± 4.33	78.90	49.27	43.61	68.69 ± 3.37	84.38	50.58	45.03

Table 4.10. Performance of coupled and parallel HMMs after DTW with respect to accuracy, precision, recall and overlap with the enlarged window $(+21, +6)$.

Feature sets		DTW-cHMM $(+21, +6)$				DTW-pHMM $(+21, +6)$			
HMM 1	HMM 2	Acc	Pre	Rec	Ovr	Acc	Pre	Rec	Ovr
CoM	E	82.68 ± 2.38	79.21	71.08	60.28	83.10 ± 2.11	79.44	74.56	62.34
CoM	RDF	81.57 ± 1.76	74.76	79.50	62.40	81.61 ± 2.00	75.02	81.16	63.35
CoM	H	80.85 ± 2.28	78.02	68.26	56.55	80.57 ± 2.58	76.30	69.57	56.76
CoM	DCT	76.81 ± 3.17	67.47	67.14	50.47	77.32 ± 3.25	69.54	67.98	51.35
CoM	HOG	77.08 ± 3.37	69.45	65.20	49.86	77.78 ± 3.19	70.45	67.87	51.70
CoM	LBP	73.29 ± 2.36	66.16	46.54	36.88	73.99 ± 2.35	69.23	50.05	39.22
CoM, $\triangle$CoM	E	$\mathbf{83.44 \pm 1.98}$	81.22	73.71	62.62	$\mathbf{84.15 \pm 1.46}$	82.16	74.90	64.11
CoM, $\triangle$CoM	RDF	82.53 ± 2.07	76.14	82.76	64.91	83.54 ± 1.48	78.49	82.85	66.39
CoM, $\triangle$CoM	H	82.53 ± 1.86	81.07	70.95	60.37	81.80 ± 2.12	80.35	69.38	58.38
CoM, $\triangle$CoM	DCT	77.19 ± 3.55	67.94	67.29	50.65	78.48 ± 3.26	71.85	68.96	53.17
CoM, $\triangle$CoM	HOG	77.40 ± 3.19	70.09	66.07	50.37	79.43 ± 2.84	73.77	68.78	54.17
CoM, $\triangle$CoM	LBP	74.63 ± 2.91	66.75	51.13	40.35	74.93 ± 2.61	71.40	50.27	40.64
CoM, $\triangle$CoM, E	RDF	82.43 ± 2.10	75.54	82.85	64.63	83.96 ± 2.03	78.24	82.99	66.89
CoM, $\triangle$CoM, E	H	82.48 ± 1.86	79.37	72.77	60.75	81.62 ± 2.25	78.83	70.40	58.35
CoM, $\triangle$CoM, E	DCT	77.84 ± 3.05	69.26	68.17	51.68	78.94 ± 3.37	71.92	69.48	53.62
CoM, $\triangle$CoM, E	HOG	77.86 ± 3.31	69.38	67.34	51.56	79.36 ± 2.77	73.44	68.55	53.92
CoM, $\triangle$CoM, E	LBP	74.41 ± 2.98	68.16	49.44	39.44	75.17 ± 2.64	71.93	50.75	41.13

Table 4.11. The effect of occlusion, and number of hands involved in signing.

	Accuracies of DTW-HMM $(+21, +6)$				
Feature sets	overall	one-hand	two-hands	occlusion	no occlusion
CoM	82.56	81.35	83.77	82.24	82.77
△CoM	78.69	77.60	79.78	78.47	78.83
E	82.07	80.55	83.58	82.42	81.83
RDF	77.58	78.64	76.51	76.34	78.40
H	78.41	77.12	79.70	78.76	78.18
DCT	73.20	72.23	74.18	73.62	72.92
HOG	74.04	72.65	75.44	74.36	73.83
LBP	71.09	70.37	71.80	70.77	71.30
CoM, △CoM	84.19	83.55	84.84	83.20	84.85
CoM, E	84.25	83.18	85.32	84.37	84.17
CoM, RDF	81.36	81.95	80.76	80.21	82.12
CoM, H	83.43	82.52	84.34	83.95	83.09
CoM, DCT	77.11	77.32	76.90	77.74	76.69
CoM, HOG	77.02	76.72	77.32	78.19	76.23
CoM, LBP	74.57	74.34	74.80	74.89	74.35
CoM, △CoM, E	84.61	83.71	85.51	84.28	84.83
CoM, △CoM, RDF	80.98	81.54	80.42	80.05	81.59
CoM, △CoM, H	83.43	82.62	84.24	83.13	83.63
CoM, △CoM, DCT	78.36	78.49	78.23	78.70	78.14
CoM, △CoM, HOG	77.57	77.46	77.68	78.20	77.14
CoM, △CoM, LBP	74.93	74.92	74.94	75.43	74.59
CoM, △CoM, E, RDF	82.37	82.01	82.74	82.28	82.43
CoM, △CoM, E, H	83.77	82.87	84.66	83.92	83.66
CoM, △CoM, E, DCT	78.83	78.14	79.52	78.84	78.83
CoM, △CoM, E, HOG	78.07	77.19	78.95	78.77	77.60
CoM, △CoM, E, LBP	75.04	74.43	75.65	75.30	74.87

Table 4.12. The effect of duration.

	Accuracies of DTW $(+21, +6)$			Accuracies of DTW-HMM $(+21, +6)$		
Feature sets	overall	short signs	long signs	overall	short signs	long signs
CoM	78.40	75.96	82.05	82.56	82.44	82.74
△CoM	78.60	78.56	78.67	78.69	80.05	76.65
E	78.60	75.74	82.90	82.07	81.47	82.95
RDF	73.41	71.19	76.74	77.58	76.65	78.96
H	78.81	76.72	81.95	78.41	78.02	79.00
DCT	72.73	69.73	77.24	73.20	72.47	74.31
HOG	73.56	70.63	77.94	74.04	72.81	75.89
LBP	73.03	69.82	77.85	71.09	71.27	70.81
CoM, △CoM	82.40	81.95	83.08	84.19	85.62	82.05
CoM, E	79.45	76.84	83.36	84.25	83.93	84.73
CoM, RDF	76.68	75.04	79.14	81.36	81.04	81.83
CoM, H	78.80	76.08	82.88	83.43	82.97	84.12
CoM, DCT	76.49	73.75	80.60	77.11	76.48	78.06
CoM, HOG	76.50	73.45	81.08	77.02	76.12	78.36
CoM, LBP	76.33	73.44	80.68	74.57	74.52	74.63
CoM, △CoM, E	79.91	77.79	83.10	84.61	84.63	84.58
CoM, △CoM, RDF	76.24	74.39	79.03	80.98	80.38	81.87
CoM, △CoM, H	79.79	77.74	82.87	83.43	83.45	83.40
CoM, △CoM, DCT	78.25	75.70	82.09	78.36	78.06	78.82
CoM, △CoM, HOG	77.73	74.83	82.07	77.57	76.93	78.52
CoM, △CoM, LBP	77.56	74.87	81.60	74.93	75.35	74.29
CoM, △CoM, E, RDF	77.44	75.17	80.85	82.37	81.70	83.37
CoM, △CoM, E, H	78.83	76.39	82.49	83.77	83.60	84.01
CoM, △CoM, E, DCT	77.88	75.16	81.96	78.83	78.43	79.44
CoM, △CoM, E, HOG	77.61	74.80	81.83	78.07	77.65	78.69
CoM, △CoM, E, LBP	77.20	74.33	81.51	75.04	75.23	74.75

Table 4.13. Recognition performances for the enlarged intervals $(+21, +6)$, ground truth intervals, intervals found by DTW and DTW-HMM.

	Recognition Performances (%)			
Feature sets	$(+21, +6)$	GT	DTW	DTW-HMM
CoM	12.00 ± 5.47	35.25 ± 7.81	28.42 ± 6.35	34.67 ± 6.01
△CoM	25.83 ± 5.43	60.42 ± 7.69	51.17 ± 7.30	52.67 ± 8.86
E	12.00 ± 5.10	36.67 ± 7.20	26.42 ± 7.00	35.58 ± 6.72
RDF	93.25 ± 4.00	95.67 ± 3.77	95.25 ± 3.62	95.08 ± 4.02
H	7.67 ± 3.71	17.17 ± 5.20	14.25 ± 5.91	15.50 ± 5.43
DCT	90.58 ± 7.33	90.67 ± 6.73	90.67 ± 7.37	90.67 ± 7.16
HOG	88.67 ± 9.32	88.83 ± 9.00	88.58 ± 8.85	88.00 ± 8.60
LBP	5.42 ± 3.22	6.83 ± 4.25	4.83 ± 3.28	10.92 ± 4.57
CoM, △CoM	51.83 ± 8.25	81.83 ± 4.91	76.08 ± 5.11	81.58 ± 5.39
CoM, E	28.00 ± 6.14	63.25 ± 8.96	54.33 ± 7.16	63.33 ± 6.54
CoM, RDF	96.00 ± 3.39	97.25 ± 2.89	96.58 ± 2.67	97.17 ± 2.84
CoM, H	15.00 ± 5.76	43.08 ± 6.11	35.08 ± 6.99	44.25 ± 7.46
CoM, DCT	90.83 ± 6.96	91.25 ± 7.45	90.58 ± 7.36	90.75 ± 8.20
CoM, HOG	88.83 ± 8.95	88.50 ± 8.90	89.00 ± 9.14	89.25 ± 8.31
CoM, LBP	9.00 ± 5.67	24.67 ± 5.28	15.25 ± 4.75	24.75 ± 6.74
CoM, △CoM, E	52.33 ± 6.60	86.42 ± 5.24	77.42 ± 7.18	85.42 ± 5.50
CoM, △CoM, RDF	$\mathbf{97.17 \pm 2.69}$	$\mathbf{98.25 \pm 2.19}$	$\mathbf{98.33 \pm 1.78}$	$\mathbf{98.50 \pm 1.93}$
CoM, △CoM, H	27.08 ± 6.06	63.33 ± 7.05	54.08 ± 7.12	65.25 ± 7.17
CoM, △CoM, DCT	90.92 ± 6.77	90.83 ± 7.29	90.92 ± 7.50	90.67 ± 7.88
CoM, △CoM, HOG	88.67 ± 8.97	89.08 ± 8.80	88.92 ± 8.92	88.33 ± 8.57
CoM, △CoM, LBP	13.50 ± 3.05	37.25 ± 8.99	27.50 ± 4.69	34.42 ± 7.84
CoM, △CoM, E, RDF	$\mathbf{97.17 \pm 2.99}$	$\mathbf{98.17 \pm 2.45}$	$\mathbf{97.75 \pm 3.10}$	$\mathbf{98.67 \pm 1.70}$
CoM, △CoM, E, H	27.92 ± 6.27	66.42 ± 7.33	55.83 ± 5.99	68.17 ± 7.04
CoM, △CoM, E, DCT	91.00 ± 7.12	91.08 ± 7.30	90.83 ± 6.86	91.42 ± 6.88
CoM, △CoM, E, HOG	89.25 ± 8.74	89.42 ± 9.32	89.17 ± 9.06	89.33 ± 8.98
CoM, △CoM, E, LBP	15.92 ± 4.66	45.58 ± 8.30	32.33 ± 7.16	38.42 ± 7.12

5. CONCLUSION

In this study, we investigate multiple sequence alignment methods for extraction of isolated signs in continuous sign language videos. For this task we are dealing with preprocessing of the hand images, feature extraction to represent the hand gesture, and alignment of the videos.

Several feature extraction methods are tested to describe hand gestures both for segmentation and recognition tasks. These features can be categorized as follows: Center of mass coordinates of each hand and their first-order derivatives, Ellipse parameters for each hand, Discrete Cosine Transform, Histogram of oriented gradients, Local Binary Patterns, Hu Moments, and Radial Distances with respect to a reference point around the center of the hand.

The contribution of this thesis is to perform direct sign segmentation, without the need for a pre-trained sign model. Using dynamic time warping and hidden Markov model variant methods such as continuous left-to-right HMMs, coupled HMMs and parallel HMMs, we align sign video sequences that contain the same information to extract the longest common part in these videos. We further investigate performances of recognition based on continuous HMMs to compare feature extraction methods.

The proposed system is tested on a database consisting of Turkish signed speech videos of TRT broadcast news for the hearing impaired. In this database we have 15 videos with 17,4939 frames and a total of 10,318 words which correspond to 3,498 different signs. In all of the videos, the same newscaster is presenting the news by speaking and signing simultaneously. Moreover a subtitle presenting the same information as the newscaster accompanies the video. The exact start and end locations of the signs are manually annotated by TSL signers.

For our experiments we selected a subset of this database among the most frequent words. This subset contains 1,200 sign samples in total, and consists of 40 words, where

for each word there are 30 sign samples. For 20 of the selected words, the corresponding signs are one handed and for the other 20, the signs are two handed. Moreover, in 47.83 per cent of the samples occlusion is observed. Having nearly half of the signs with occlusion or contact, we can describe our database as a challenging one.

In our experiments we have seen that for the segmentation task, the main information is contained in motion and speed of the hands. Moreover, simple shape descriptors such as ellipse parameters, RDF or Hu moments give better results compared to high level shape descriptors.

When the synchronization of the speech and the sign is high, DTW performs better than HMM variant methods. When the synchronization is poor, the erroneous instances can be improved by the use of HMM. However, the improvement in the performance is negligible considering that DTW can be applied for a real time setting, whereas training of HMM has a high time complexity. We observed that training of DTW for 1,200 sign sequences can be achieved in several seconds whereas duration of training the HMM for the same amount of data is about two hours.

We apply our algorithm to the sign sequences, where the search interval is found by the speech recognizer and enlarged using the parameters $(+15, -1)$ and $(+21, +6)$. When we align the sequences using DTW, the best performance is obtained using center of mass features and their first order derivatives. The best accuracy is shown to be 79.69 per cent in a search interval that is enlarged by $(+15, -1)$, and 82.40 per cent in a search interval that is enlarged by $(+21, +6)$.

Using HMM alone results in poor segmentation when compared to DTW. Using coupled HMM or parallel HMM for fusing the features improves the performances of HMM, however a satisfactory result is only obtained when we apply HMM to the intervals found by DTW and combine the strengths of these algorithms. In this approach, the best accuracy is obtained using the features center of mass, their first order derivatives and ellipse parameters. In this approach we obtain an accuracy of 80.45 per cent for the search interval enlarged by $(+15, -1)$, and 84.61 per cent in a search interval

that is enlarged by $(+21, +6)$.

When we apply recognition, we see that the performance obtained by using motion, speed or simple shape descriptors is consistently low. In this task, high level shape descriptors are needed to represent hand gestures. Surprisingly, the best performance is obtained when RDF features are used to describe shape in combination with motion and speed information, where we achieved 98.33 per cent recognition accuracy.

The most important observation is that when we use the intervals by automatically segmenting the signs, we obtain better performance compared to manually annotated intervals. Therefore we can conclude that our segmentation is accurate.

In a real setting, it is not realistic to find 30 samples for a given word. In fact, we should expect to find at most 10 samples. Moreover, it may be the case that the samples not always contain the same information due to the error made in speech recognition or homonymic words. Therefore, in the future we are aiming to apply clustering before segmenting the samples. In clustering we will combine the results obtained by segmentation and recognition.

REFERENCES

1. Ong, S. C. and S. Ranganath, "Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 6, pp. 873–891, June 2005.

2. Vogler, C. and D. Metaxas, "Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 156–161, 1997.

3. Vogler, C. and D. Metaxas, "Parallel hidden Markov models for American sign language recognition", *International Conference on Computer Vision*, Vol. 1, pp. 116–122, 1999.

4. Fang, G. and W. Gao, "A SRN/HMM System for Signer-independent Continuous Sign Language Recognition", *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

5. Fang, G., W. Geo, and D. Zhao, "Large-Vocabulary Continuous Sign Language Recognition Based in Transition-Movement Models", *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, Vol. 37, No. 1, pp. 1–9, January 2007.

6. Kong, W. and S. Ranganath, "Signing Exact English (SEE): Modeling and Recognition", *Pattern Recognition*, Vol. 41, No. 5, pp. 1655–1669, May 2008.

7. Aran, O., I. Ari, A. Benoit, P. Campr, A. H. Carrillo, F.-X. Fanard, L. Akarun, A. Caplier, , and B. Sankur, "Signtutor: an interactive system for sign language tutoring", *IEEE Multimedia*, Vol. 16, 2009.

8. Brand, M., "Coupled Hidden Markov Models for Modeling Interacting Processes", Technical report, MIT Media Lab Perceptual Computing, June 1997.

9. Lichtenauer, J. F., E. A. Hendriks, and M. J. Reinders, "Sign Language Recognition by Combining Statistical DTW and Independent Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 11, pp. 2040–2046, November 2008.

10. Starner, T., A. Pentland, and J. Weaver, "Real-time American Sign Language recognition using desk and wearable computer based video", *IEEE Transactions on Pattern Analysis And Machine Intelligence*, Vol. 20, No. 12, pp. 1371–1375, December 1998.

11. Lee, H. K. and J. H. Kim, "An HMM-Based Threshold Model Approach For Gesture Recognition", *IEEE Transactions on Pattern Analysis And Machine Intelligence*, Vol. 21, No. 10, pp. 961–973, October 1999.

12. Corradini, A., "Dynamic Time Warping for Off-line Recognition of a Small Gesture Vocabulary", *Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, pp. 82–89, 2001.

13. Liu, N., B. C. Lovell, P. J. Kootsookos, and R. I. Davis, "Model structure selection and training algorithms for an HMM gesture recognition system", *IWFHR '04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, pp. 100–105, 2004.

14. Nayak, S., S. Sarkar, and B. Loeding, "Unsupervised Modeling of Signs Embedded in Continuous Sentences", *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pp. 81–88, 2005.

15. Nayak, S., S. Sarkar, and B. Loeding, "Distribution-based dimensionality reduction applied to articulated motion recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 5, pp. 795–810, 2009.

16. Alon, J., V. Athitsos, Q. Yuan, 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.

17. von Agris, U., J. Zieren, U. Canzler, B. Bauer, and K.-F. Kraiss, "Recent developments in visual sign language recognition", *Universal Access in the Information Society*, Vol. 6, No. 4, pp. 323–362, 2008.

18. Aran, O., I. Ari, P. Campr, E. Dikici, M. Hruz, S. Parlak, L. Akarun, and M. Saraclar, "Speech and sliding text aided sign retrieval from hearing impaired sign news videos", *Journal on Multimodal User Interfaces*, Vol. 2, No. 2, pp. 117–131, 2008.

19. Aran, O., *Vision based sign language recognition: modeling and recognizing isolated signs with manual and non-manual components*, Ph.D. thesis, Bogazici University, 2008.

20. Ari, I., *Facial Feature Tracking and Expression Recognition for Sign Language*, Master's thesis, Bogazici University, 2008.

21. von Agris, U., M. Knorr, and K. F. Kraiss, "The significance of facial features for automatic sign language recognition", *8th IEEE International Conference on Automatic Face and Gesture Recognition*, September 2008.

22. Ekenel, H. K., M. Fischer, E. Tekeli, R. Stiefelhagen, and A. Ercil, "Local binary pattern domain local appearance face recognition", *IEEE 16th Signal Processing, Communication and Applications Conference (SIU08)*, pp. 1–4, 2008.

23. Shan, C., S. Gong, and P. W. McOwan, "Facial expression recognition based on Local Binary Patterns: A comprehensive study", *Image and Vision Computing*, Vol. 27, No. 6, pp. 803–816, 2009.

24. Just, A., Y. Rodriguez, and S. Marcel, "Hand Posture Classification and Recognition using the Modified Census Transform", *Proceedings of the 7th International*

Conference on Automatic Face and Gesture Recognition, pp. 351–356, 2006.

25. Freeman, W. T. and M. Roth, "Orientation histograms for hand gesture recognition", *In International Workshop on Automatic Face and Gesture Recognition*, pp. 296–301, 1995.

26. Binh, N. D., E. Shuichi, and T. Ejima, "Real-Time Hand Tracking and Gesture Recognition System", *Proceedings of International Conference on Graphics, Vision and Image Processing (GVIP-05)*, pp. 362–368, 2005.

27. Farhadi, A. and D. Forsyth, "Aligning ASL for Statistical Translation Using a Discriminative Word Model", *Computer Vision and Pattern Recognition*, Vol. 2, pp. 1471–1476, 2006, `http://dx.doi.org/10.1109/CVPR.2006.51`.

28. Brand, M., N. Oliver, and A. Pentland, "Coupled Hidden Markov Models for Complex Action Recognition", *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 994–999, 1997.

29. Pavlovic, V. I., *Dynamic Bayesian Networks For Information Fusion With Applications To Human-Computer Interfaces*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1999.

30. Lichtenauer, J., E. Hendriks, and M. Reinders, "Learning to Recognize a Sign from a Single Example", *8th IEEE International Conference on Automatic Face and Gesture Recognition*, September 2008.

31. Durbin, R., S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge, U.K.: Cambridge University Press, 1998.

32. Notredame, C., "Recent progresses in multiple sequence alignment: a survey", *Pharmacogenomics*, Vol. 3, No. 1, pp. 131–144, January 2002.

33. Parizeau, M. and R. Plamondon, "A Comparative Analysis of Regional Correlation,

Dynamic Time Warping, and Skeletal Tree Matching for Signature Verification", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 12, No. 7, pp. 710–717, 1990.

34. Kholmatov, A., *Biometric Identity Verification Using On-Line & Off-Line Signature Verification*, Master's thesis, Sabanci University, July 2003.

35. Park, A. S. and J. R. Glass, "Unsupervised Pattern Discovery in Speech", *IEEE Transactions on Audio, Speech, And Language Processing*, Vol. 16, No. 1, pp. 186–197, January 2008.

36. Aran, O., T. Burger, A. Caplier, and L. Akarun, "A belief-based sequential fusion approach for fusing manual signs and non-manual signals", *Pattern Recognition*, Vol. 42, No. 5, pp. 812–822, May 2009.

37. Nefian, A. V., L. Liang, X. Pi, L. Xiaoxiang, C. Mao, and K. Murphy, "A Coupled Hmm For Audio-Visual Speech Recognition", *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, Vol. 2, pp. 2013–2016, 2002.

38. Nefian, A. V., L. Liang, X. Pi, X. Liu, and K. Murphy, "Dynamic Bayesian Networks for Audio-Visual Speech Recognition", *EURASIP Journal on Applied Signal Processing*, Vol. 11, pp. 1274–1288, 2002.

39. Cooper, H. and R. Bowden, "Learning Signs from Subtitles: A Weakly Supervised Approach to Sign Language Recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009.

40. Buehler, P., M. Everingham, and A. Zisserman, "Learning sign language by watching TV (using weakly aligned subtitles)", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009.

41. Nayak, S., S. Sarkar, and B. Loeding, "Automated Extraction of Signs from Con-

tinuous Sign Language Sentences using Iterated Conditional Modes", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009.

42. Yang, H.-D., A.-Y. Park, and S.-W. Lee, "Robust Spotting Of Key Gestures From Whole Body Motion Sequence", *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 231–236, 2006.

43. Arisoy, E., H. Sak, and M. Saraclar, "Language Modeling for Automatic Turkish Broadcast News Transcription", *Interspeech*, 2007.

44. Arisoy, E., D. Can, S. Parlak, H. Sak, and M. Saraclar, "Turkish Broadcast News Transcription and Retrieval", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. Special issue on morphologically rich languages, 2009.

45. Campr, P., M. Hruz, A. Karpov, P. Santemiz, M. Zelezny, and O. Aran, "Sign-language-enabled information kiosk", *Proceedings of the 4th International Summer Workshop on MultiModal Interfaces (eNTERFACE08)*, pp. 24–33, 2008.

46. Gonzalez, R. C. and R. E. Woods, *Digital Image Processing*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

47. Konukoglu, E., E. Yoruk, J. Darbon, and B. Sankur, "Shape-Based Hand Recognition", *IEEE Transactions on Image Processing*, Vol. 15, No. 7, pp. 1803–1815, 2006.

48. Hu, M.-K., "Visual pattern recognition by moment invariants", *IRE Transactions on Information Theory*, Vol. 8, No. 2, pp. 179–187, 1962.

49. Ekenel, H. K. and R. Stiefelhagen, "Analysis of Local Appearance-Based Face Recognition: Effects of Feature Selection and Feature Normalization", *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW06)*, 2006.

50. Dalal, N. and B. Triggs, "Histograms of Oriented Gradients for Human Detection",

In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp. 886–893, 2005.

51. Alpaydin, E., *Introduction to machine learning*, The MIT Press, 2004.

52. Igarza, J. J., I. Goirizelaia, K. Espinosa, I. Hernaez, R. Mendez, and J. Sanchez, "Online Handwritten Signature Verification Using Hidden Markov Models", *Progress in Pattern Recognition, Speech and Image Analysis*, Vol. 2905, pp. 391–399, Springer Berlin / Heidelberg, 2003.

53. Rabiner, L. R. and B. Juang, "An Introduction To Hidden Markov Models", *IEEE ASSP Magazine*, Vol. 3, No. 1, pp. 4–16, January 1986.