VISION BASED SIGN LANGUAGE RECOGNITION: MODELING AND RECOGNIZING ISOLATED SIGNS WITH MANUAL AND NON-MANUAL COMPONENTS

by

Oya Aran

B.S, in CmpE., Boğaziçi University, 2000

M.S, in CmpE., Boğaziçi University, 2002

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

Graduate Program in Computer Engineering

Boğaziçi University

2008

ACKNOWLEDGEMENTS

My first and foremost acknowledgment must go to my supervisor Prof. Lale Akarun. During the long journey of this thesis study, she supported me in every aspect. She was the one who persuaded me to continue my graduate studies and she inspired me with her enthusiasm on research, her experience, and her lively character. I learned much from her, both academically and personally. In every sense, this thesis would not have been possible without her.

I would like to thank to Prof. Bülent Sankur for his valuable ideas, comments, and creating academic opportunities, to my proffesors Fikret Gürgen, Volkan Atalay, Pınar Yolum Birbil, Ethem Alpaydın, Ali Taylan Cemgil, and Murat Saraçlar, for their feedbacks and fruitful discussions, and to TUBITAK for their support. Parts of this thesis, particularly the belief based approach in Chapter 5, would not have been complete without our collaboration with Thomas Burger and Prof. Alice Caplier. I would also like to thank to the members of the project groups at the eNTERFACE workshops that I worked together with.

I would especially like to thank to Prof. Cem Ersoy for creating an excellent and joyful atmosphere in the department and to my friends and colleagues, Neşe Alyüz, İsmail Arı, Koray Balcı, İlker Demirkol, Onur Dikmen, Berk Gokberk, Mehmet Gönen, Burak Gürdağ, Itır Karaç, Cem Keskin, Rabun Koşar, Atay Özgövde, Albert Ali Salah, Pınar Santemiz, Burak Turhan, Aydın Ulaş, and Olcay Taner Yıldız, for our scientific and philosophical discussions and for all the fun we had.

I owe special gratitude to my mother, father and brother for their continuous, unconditional love and support, and for giving me the inspiration and motivation. My last, and most heartfelt, acknowledgment must go to my husband Onur. He was always there for me, with his love, patience and support, when I needed the most.

ABSTRACT

VISION BASED SIGN LANGUAGE RECOGNITION: MODELING AND RECOGNIZING ISOLATED SIGNS WITH MANUAL AND NON-MANUAL COMPONENTS

This thesis addresses the problem of vision based sign language recognition and focuses on three main tasks to design improved techniques that increase the performance of sign language recognition systems. We first attack the markerless tracking problem during natural and unrestricted signing in less restricted environments. We propose a joint particle filter approach for tracking multiple identical objects, in our case the two hands and the face, which is robust to situations including fast movement, interactions and occlusions. Our experiments show that the proposed approach has a robust tracking performance during the challenging situations and is suitable for tracking long durations of signing with its ability of fast recovery. Second, we attack the problem of the recognition of signs that include both manual (hand gestures) and non-manual (head/body gestures) components. We investigated multi-modal fusion techniques to model the different temporal characteristics and propose a two-step sequential belief based fusion strategy. The evaluation of the proposed approach, in comparison to other state of the art fusion approaches, shows that our method models the two modalities better and achieves higher classification rates. Finally, we propose a strategy to combine generative and discriminative models to increase the sign classification accuracy. We apply the Fisher kernel method and propose a multi-class classification strategy for gesture and sign sequences. The results of the experiments show that the classification power of discriminative models and the modeling power of generative models are effectively combined with a suitable multi-class strategy. We also present two applications, a sign language tutor and an automatic sign dictionary, developed based on the ideas and methods presented in this thesis.

ÖZET

VİDEO TABANLI İŞARET DİLİ TANIMA: EL VE EL DIŞI HAREKETLER İÇEREN AYRIK İŞARETLERİN MODELLENMESİ VE TANINMASI

Bu tezde kamera tabanlı işaret dili tanıma problemi üzerine çalışılmış ve üç alt problemde yoğunlaşılmıştır: (1) belirteçsiz el izleme, (2) çok kipli tümleştirme, (3) tanıma. Bu alt problemler için literatürde sunulan çalışmalara göre daha gelişmiş teknikler önerilmiş ve karşılaştırmalı analizler yapılmıştır. İşaret dilinde eller birbirini ya da yüzü kapatabilir. Bu tür durumlarda da gürbüz izleme yapabilecek bir izleme algoritmasına ihityaç vardır. Bu çalışmada çok sayıda nesnenin takibi sırasında temas ve kapatma durumlarında da gürbüz izleme yapabilen, birleşik parçacık süzgeci tabanlı bir yöntem önerdik. Yapılan testlerde önerilen yöntemin temas ve kapatmaya karşı gürbüz olduğu ve mevcut yöntemlere göre daha iyi çalıştığı gözlendi. İşaret dili, temelinde el hareketleri ve el şekline dayanan fakat bunların yanında yüz mimiklerinin, baş ve vücut hareketlerinin de kullanıldığı görsel bir dildir. Bu çalışmada işaretlerin bu çok kipli yapısını dikkate aldık ve ardışık tümleştirme yöntemi ile inanç tabanlı bir tanıma sistemi geliştirdik. Sonuçlar önerdiğimiz yöntemin literatürdeki diğer tümleştirme yöntemlerine göre daha başarılı olduğunu gösterdi. Bu çalışmada önerdiğimiz bir diğer yöntem ise, üretici ve ayırıcı modellerin birleştirilerek işaret tanıma amaçlı kullanılması üzerinedir. İşaret tanıma probleminde yoğunlukla kullanılan üretici modelleri, ayırıcı modellerin sınıflandırma gücü ile birleştirmek için Fisher çekirdeklerini kullandık ve çok sınıflı sınıflandırma yöntemi önerdik. Deneylerde bu yöntemin üretici ve ayırıcı modellerin güçlü yanlarını tek bir modelde toplayarak sınıflandırma başarısını arttırdığı görülmektedir. Bu tez kapsamında ayrıca, çalışmada önerilen yöntemleri ve fikirleri kullanan iki uygulama, işaret dili eğitmeni ve otomatik işaret dili sözlüğü, geliştirilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

a_{ij}	Transition probability from state i to state j
a_t^n	First ellipse axis of the n^{th} particle at time t
$b_i(O_t)$	Observation probability in state i at time t
b_t^n	Second ellipse axis of the n^{th} particle at time t
c_t	Scaling coefficient
$f(.)$	State transition function
$h(.)$	Measurement function
I	Fisher information matrix
K	Number of classes
$m(.)$	Belief function
$m_{\copyright}$	Combined belief function
m_{ij}	Elementary belief function
M	Number of Gaussian components
N	Number of particles
T_{hg}	Model of hand/arm motion
T_{vg}	Model visual images given hand/arm motion
T_{vh}	Model visual images given gesture concept
U_x	Fisher score
$v_{x_t}^n$	x velocity of the center of the n^{th} particle at time t
$v_{y_t}^n$	y velocity of the center of the n^{th} particle at time t
w_{im}	Weight of the Gaussian component m at state i
$\mathbf{x}_0$	Initial system state
$\mathbf{x}_t$	System state at time t
$\hat{\mathbf{x}}_t$	Estimate at time t
$\mathbf{x}_t^n$	State vector of the n^{th} particle at time t
x_t^n	x coordinate of the center of the n^{th} particle at time t
y_t^n	y coordinate of the center of the n^{th} particle at time t
$\mathbf{Z}_{1:t}$	Observations from time 1 to t
$\mathbf{z}_t^n$	Joint likelihood of the n^{th} particle at time t

$z_t^{n,i}$	Likelihood of a single object, i, for the n^{th} particle
$\alpha_t(i)$	Forward variable
$\beta_t(i)$	Backward variable
γ	Uncertainty threshold
$\gamma_i(t)$	Probability of being in state i at time t
θ_t^n	Ellipse angle of the n^{th} particle at time t
μ_{im}	Mean of the Gaussian component m at state i
π_i	Prior probability of state i
π_t^n	Weight of the n^{th} particle at time t
σ	Standard deviation of the hesitation distribution
Σ_{im}	Covariance of the Gaussian component m at state i
$\phi(.)$	Mapping function
Φ_d	Threshold for distance
Φ_m	Threshold for facial movement
Φ_p	Threshold for particle weights
Ω	Frame
2^Ω	Power set
ASL	American sign language
BSL	British sign language
CCR	Correct classification rate
CoM	Center of mass
CSL	Chineese sign language
CSS	Curvature scale space
CV	Cross validation
DBN	Dynamic Bayesian networks
DTW	Dynamic time warping
EBF	Elemantary belief function
EER	Equal error rate
FA	False accept

FFR	Frames for recovery
FR	False reject
FSM	Finite state machines
FSS	Fisher score space
HCI	Human computer interaction
HMM	Hidden Markov model
ICA	Independent component analysis
LDA	Linear discriminant analysis
LRSS	Likelihood ratio score space
LSS	Likelihood score space
MRF	Markov random field
MS	Manual signs
MSh	Mean shift
NMS	Non-manual signals
PCA	Principal component analysis
PF	Particle filter
PPT	Partial pignistic transform
PT	Pignistic transform
SFBS	Sequential floating backward search
SFFS	Sequential floating forward search
SLR	Sign language recognition
SRN	Simple recurrent network
STD	Spoken term detection
STR	Sliding text recognition
SVM	Support vector machines
TDNN	Time delay neural networks
TID	Turkish sign language
TEF	Total erroneous frames
TMM	Transition movement models
WL	Wrong location

1. Introduction

Recent developments in sensor technology, including the developments in the camera hardware and commercialization of web cameras, drive the research in Human-Computer Interaction (HCI) to equip machines with means of communication that are naturally used between humans, such as speech and gestures. Although research on using the speech modality to communicate with computers is more mature, studies on using the gesture modality is relatively new. Hand gestures can be considered as both an independent way of communication and also a complementary modality to speech. Gestures are consciously and unconsciously used in every aspect of human communication and they form the basis of sign languages, the natural media of hearing-impaired communication.

Sign languages, like the spoken languages, emerge and evolve naturally within hearing-impaired communities. Within each country or region, wherever hearing-impaired communities exist, sign languages develop, independently from the spoken language of the region. Each sign language has its own grammar and rules, with a common property that they are all visually perceived.

Sign language recognition (SLR) is a multidisciplinary research area involving pattern recognition, computer vision, natural language processing and linguistics. It is a multifaceted problem not only because of the complexity of the visual analysis of hand gestures but also due to the highly multimodal nature of sign languages. Although sign languages are well-structured languages with a phonology, morphology, syntax and grammar, they are different from spoken languages: The structure of a spoken language makes use of words sequentially, whereas a sign language makes use of several body movements in parallel. The linguistic characteristics of sign language are different from those of spoken languages due to the existence of several components affecting the context, such as the use of facial expressions and head movements in addition to the hand movements.

The research on hand gesture and sign recognition has two main dimensions: *isolated* and *continuous* recognition. Isolated recognition focuses on a single hand gesture that is performed by the user and attempts to recognize it. In continuous recognition, user is expected to perform gestures one after the other and the aim is to recognize every gesture that the user performs. The continuous recognition problem is slightly different for hand gesture recognition and sign language recognition systems. In hand gesture controlled environments, the problem can be considered as a gesture spotting problem, where the task is to differentiate the meaningful gestures of the user from the unrelated ones. In sign language recognition, the continuous recognition problem includes the co-articulation problem. The preceding sign affects the succeeding one, which complicates the recognition task as the transitions between the signs should be explicitly modeled and incorporated to the recognition system. Moreover, language models are used to be able to perform on large-vocabulary databases.

Research on hand gesture analysis and recognition started with *instrumented gloves* with several sensors and trackers. Although these gloves provide accurate data for hand position and finger configuration, they require users to wear cumbersome devices on their hands. *Vision-based* systems on the other hand, provide a natural environment for users. They also introduce several challenges, such as the detection and segmentation of the hand and finger configuration, or handling occlusion. To overcome some of these challenges, several markers are used in vision based systems such as different colored gloves on each hand or colored markers on each finger. Despite the numerous studies in the literature, the problem of marker-free hand detection, segmentation, and tracking, in unrestricted environments, is still a challenging problem.

Another challenging problem of dynamic hand gestures is the motion representation problem. This has two aspects connected to each other: First, how to represent the features for a single frame; second, how to represent and model the complete motion trajectory. Most of the studies in the literature use *generative* models with their power to handle variable-length data, as the dynamical gestures produce variable-length sequences. Designing novel generative models, and integrating *discriminative* approaches to existing models which can better represent the inherent dynamics of the gestures

while increasing system performance is a next step in hand gesture recognition.

1.1. Research Overview and Contributions

In this thesis, we concentrated on hand gesture and sign language recognition and investigated each fundamental step in a vision based isolated recognition system, including tracking, multi-modal integration and recognition. Although our focus is more on sign language recognition, the proposed techniques in this thesis have a potential to be applied to different problems that focus on motion modeling and recognition, such as human action analysis and social interaction analysis.

- *Joint particle filter based tracking of hands during natural signing:* Accurate tracking of hands and face during unrestricted, natural signing is crucial for a sign language recognition system as the feature extraction quality and classification accuracy directly depend on the tracking. The challenging part of hand tracking during signing is mainly related to the fast and non-linear movement of the hand which makes the dynamic modeling very hard. Moreover, the occlusions of the hands with each other and with the face is very frequent. We state the problem as tracking multiple identical objects and present two particle filter based approaches that are robust to occlusions, and fast, non-linear movements: semi-independent and joint particle filters [1].
- *Fusion of manual and non-manual signs:* Although manual signs are considered as the main elements of the signs, a complete understanding of the sign language is not possible without analyzing non-manual signals, in the form of head movements, body posture and facial expressions, and integrating them into the recognition system. The different temporal characteristics of the manual signs and the non-manual signals make the multi-modal integration problem even harder. We present two sequential fusion methods based on Hidden Markov Models (HMMs) for the multi-modal fusion of manual and non-manual components. The fusion is handled by the use of HMM likelihoods in the first method, whereas the second method uses belief based decision making [2, 3, 4, 5, 6, 7].
- *Combining discriminative and generative models for multi-class classification of*

hand gestures: The great variability in gestures and signs, both in time, size, and position, as well as interpersonal differences, makes the recognition task difficult. With their power in modeling sequence data and processing variable length sequences, modeling hand gestures using generative models, such as HMMs, is a natural extension. On the other hand, discriminative methods such as Support Vector Machines (SVM), have flexible decision boundaries and better classification performance. By extracting features from gesture sequences via Fisher Kernels based on HMMs, classification can be done by a discriminative classifier. We applied the Fisher kernel strategy to combine the generative and discriminative classifiers for recognizing gesture and sign sequences. We present a multi-class classification strategy on Fisher score spaces and evaluate the performance via comparisons with state of the art multi-class classification strategies [8, 9].

- *Prototype applications:* With the framework of this thesis, we have designed two applications that use the ideas presented in this thesis. We have designed and developed an interactive system, called SignTutor, for tutoring signs to novice students [2]. The system provides an automatic evaluation of the signing of the student, and gives feedback about the quality of the manual signing, as well as the non-manual signing. The second application, Signiary, is an automatically created sign dictionary, where the users write a word in a text box and the application automatically retrieves examples of sign videos of the query word from a huge collection of videos recorded from the Turkish broadcast news for the hearing impaired [10]. Tracking, sign alignment and sign clustering techniques are applied to cluster the retrieved videos to show homonyms or pronunciation differences.

1.2. Thesis Outline

Chapter 2 is an introduction to the types and characteristics of the gestures used in human communication, including hearing impaired communication, and human computer interaction systems. Chapter 3 presents the review of state-of-the-art hand gesture and sign language recognition systems. Contributions of the thesis are presented in Chapters 4-7.

Chapter 4 presents the particle filter based tracking algorithms for marker-free hands and face tracking during signing. Two methods are presented: a joint particle filter and a semi-independent particle filter.

Isolated sign recognition, for signs containing both manual and non-manual components is explained in Chapter 5. This chapter presents two sequential fusion methods for fusing manual and non-manual signs, based on likelihoods and belief functions, respectively.

In Chapter 6, we present a hand gesture and sign classification model, which is a combination of generative and discriminative models.

Chapter 7 introduces two applications that are developed during this thesis. An interactive sign tutoring system, SignTutor and an automatically created sign dictionary, SIGNIARY.

Although each chapter has a separate conclusion section, in Chapter 8, we present an overall conclusion and discussion of the contributions of the thesis.

2. Hand Gestures In Human Communication

Websters Dictionary defines a gesture as, (1) “a movement usually of the body or limbs that expresses or emphasizes an idea, sentiment, or attitude”; (2) “the use of motions of the limbs or body as a means of expression”. Most of the gestures are performed with the hand but also with the face and the body. In the case of hand gestures, the shape of the hand, together with its movement and position with respect to the other body parts, forms a hand gesture. Gestures are used in many aspects of human communication. They can be used to accompany speech, or alone for communication in noisy environments or in places where it is not possible to talk. In a more structured way, they are used to form the sign languages of the hearing-impaired people. With the progress on HCI, the gestures have found a new area of usage. Systems that enable the use of computer programs with hand gestures, such as operating system control, games, and virtual reality applications have been developed.

2.1. Hand Gestures Accompanying Speech

Hand gestures are frequently used in human to human communication either alone or together with speech. There is considerable evidence that hand gestures are produced unconsciously along with speech in many situations and enhance the content of accompanying speech. It is also known that even when the listener can not see the hands of the speaker or there is no listener at all, hand gestures are produced.

Although hand gesture recognition for HCI is a relatively recent research area, the research on hand gestures for human-human communication is well developed. Several taxonomies are presented in the literature by considering different aspects of gestures. Gestures can be classified with respect to their independence such as autonomous gestures (independent gestures) and gesticulation (gestures that are used together with another means of communication) [11]. In [12], gestures are classified into three groups: iconic gestures, metaphoric gestures and beats. In [13], gestures are classified into four groups: conversational, controlling, manipulative and communicative. A similar cate-

gorization is given in [14, 15], which views the gestures in terms of the relation between the intended interpretation and the abstraction of the movement. The conversational and controlling gesture types in [16] are accepted as a sub category under communicative gestures. In [17], this taxonomy of gestures is accepted as the most appropriate one for HCI purposes. An extended version of this taxonomy is given in Figure 2.1.

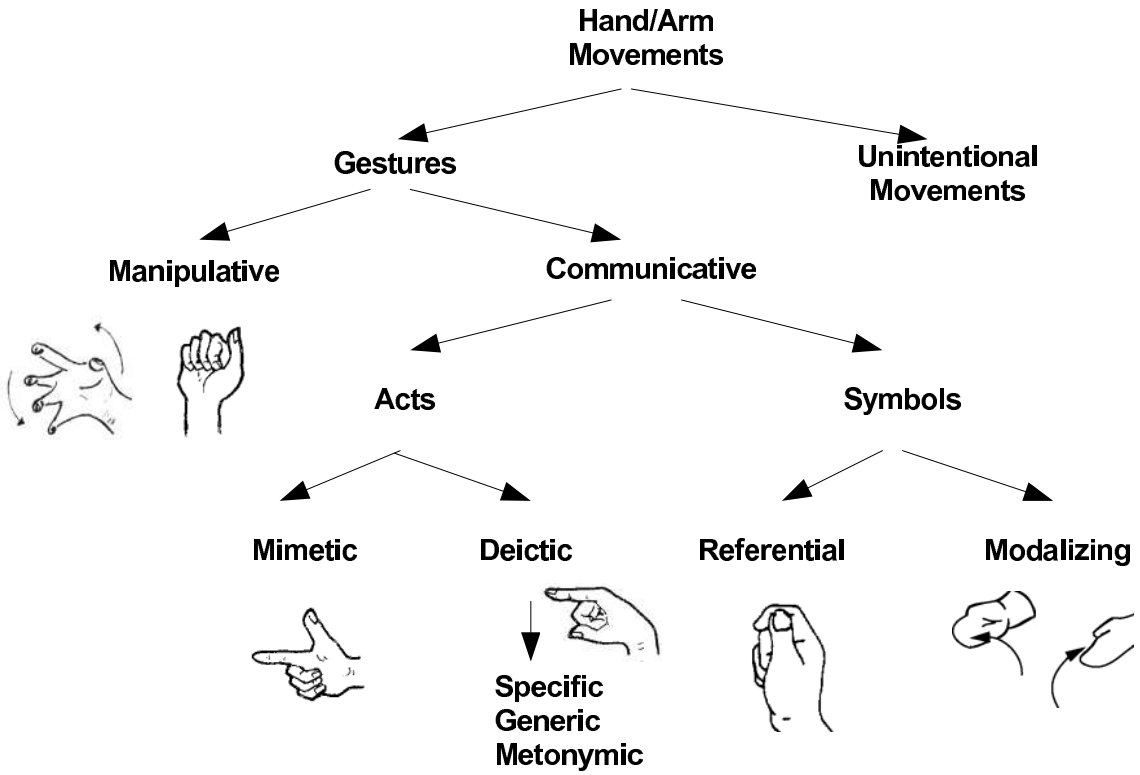


Figure 2.1. A taxonomy of hand gestures for HCI [17]

The hand/arm movements during conversation can generally be classified into two groups: intended or unintended. Although unintended hand movements must also be taken into account in order to realize human-computer interaction as natural as human-human interaction, current research on gesture recognition focuses on intended gestures which are used for either communication or manipulation purposes. *Manipulative* gestures are used to act on objects; such as rotation, grasping, etc. *Communicative* gestures have an inherent communicational purpose. In a natural environment they are usually accompanied by speech.

Communicative gestures can be *acts* or *symbols*. Symbol gestures are generally used in a linguistic role with a short motion (i.e. sign languages). In most cases, the symbol itself is not directly related to the meaning and these gestures have a predetermined convention. Two types of symbol gestures are *referential* and *modalizing*. Referential gestures are used to refer to an object or a concept independently. For example, rubbing the index finger and the thumb in a circular fashion independently refers to money. Modalizing gestures are used with some other means of communication, such as speech. For example, the sentence "I saw a fish. It was this big." is only meaningful with the gesture of the speaker. Another example is the symbol for continuation which means that the person should continue to whatever he/she is doing. Unlike symbol gestures, act gestures are directly related to the intended interpretation. Such movements are classified as either *mimetic* or *deictic*. Mimetic gestures usually mimic the concept or object to refer. For example, a smoker going through the motion of "lighting up" with a cigarette in his mouth indicates that he needs a light, or hand is like holding a gun. Deictic gestures or pointing gestures are used for pointing to objects and classified as specific, generic or metonymic by its context.

There is another type of gesture that have different characteristics than the other gesture types. These are called *beat* gestures [18, 12]. In beat gestures, the hand moves, in a short and quick way, along with the rhythm of speech. A typical beat gesture is a simple and short motion of the hand or the fingers up and down or back and forth. The main effect of a beat gesture is that it emphasizes the phrase it accompanies.

In order to differentiate intended and unintended hand/arm movements or different gestures, one should know the exact start and end time of a gesture. This is called the gesture segmentation problem. Gestures are dynamic processes and the temporal characteristics of gestures are important for segmentation purposes. With the exception of beat gestures, each gesture starts, continues for some interval and ends. This is not only valid for dynamic gestures that include both the spatial and the temporal component, but also for static gestures that only contain the spatial component. A gesture is constituted in three phases [19, 18, 11, 12]: preparation, stroke, and retraction or recovery. In the preparation phase, the hand is oriented for the gesture. The stroke

phase is the phase of the actual gesture. Finally, in the retraction phase, the hand returns to the rest position. The preparation and the stroke phases constitute a gesture phrase and together with the recovery phase, they constitute a gesture unit [19].

2.2. Hand Gestures in Hearing Impaired Communication

Sign languages are the natural communication media of hearing-impaired people all over the world. Like the spoken languages, they emerge spontaneously, and evolve naturally within hearing-impaired communities. Wherever hearing-impaired communities exist, sign languages develop, without necessarily having a connection with the spoken language of the region. American Sign Language, British Sign Language, Turkish Sign Language, and French Sign Language are different sign languages used by corresponding communities of hearing-impaired people.

Although their history is at least as old as spoken languages, the written evidences showing sign language usage date back to the 16^{th} century in Italy, Spain or in the Ottoman courts [20]. The earliest record of sign language education dates to the 18th century: In Paris, Abbé de l'Épée founded a school to teach old French sign language and graduated Laurent Clerc who later founded the "Gallaudet College" in U.S. with T. H. Gallaudet. Gallaudet College later became Gallaudet University which is the only liberal arts university for the deaf in the world.

Sign languages are visual languages: the phonology makes use of the hand shape, place of articulation, and movement; the morphology uses directionality, aspect and numeral incorporation, and syntax uses spatial localization and agreement as well as facial expressions. The whole message is contained not only in hand motion and shapes (manual signs) but also in facial expressions, head/shoulder motion and body posture (non-manual signals). As a consequence, the language is intrinsically multimodal [21, 22].

The main difference between spoken and sign languages is the way the communicative units are produced and perceived [22]. In spoken languages, the words are

produced through the vocal tract and they are perceived as sounds; whereas in sign languages, the signs are produced alone or simultaneously, by use of hand shapes, hand motion, hand location, facial expression, head motion, and body posture, and they are perceived visually. Sign languages have both sequential and parallel nature: signs come one after the other showing a sequential behavior; however, each sign may contain parallel actions of hands, face, head or body. Apart from differences in production and perception, sign languages contain phonology, morphology, semantics, and syntax like spoken languages [21]. Figure 2.2 shows an example sign from Turkish sign language (Türk İşaret Dili - TID).

Figure 2.2. An example sign, “anne (mother)” from TID

Apart from sign languages, there are other means of hearing impaired communication: finger-spelling and cued speech.

Finger-spelling is a way to code the words with a manual alphabet which is a system of representing all the letters of an alphabet, using only the hands. Finger-spelling is a part of sign languages and is used for different purposes. It may be used to represent words which have no sign equivalent, or for emphasis, clarification, or when teaching or learning a sign language. Figure 2.3 shows the finger spelling alphabet of TID.

Cued Speech is a mode of communication based on the phonemes and properties of spoken languages. Cued Speech was developed by Dr. Cornett in 1967 [23] and it uses both lip shapes and hand gestures to represent the phonemes. The aim of

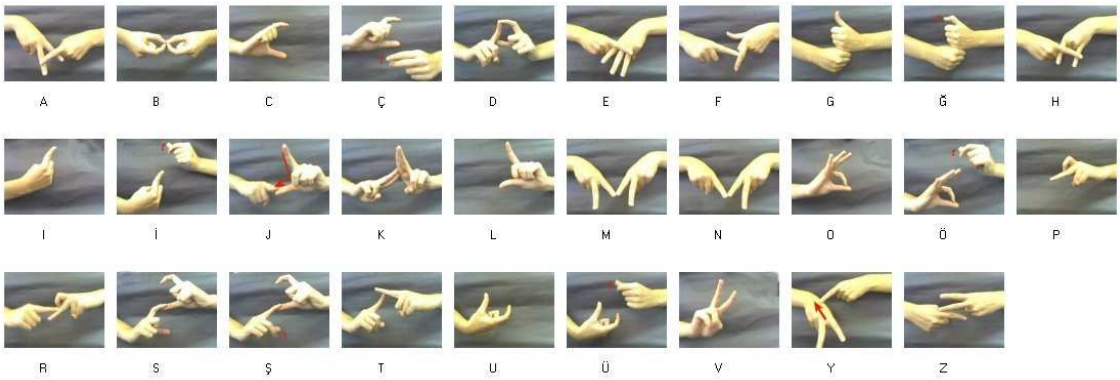


Figure 2.3. Finger-spelling alphabet of TID

cued speech is to overcome the problems of lip-reading and to enable deaf people to fully understand spoken languages. Cued speech makes the natural oral language accessible to the hearing impaired, by replacing invisible articulators that participate to the production of the sound (vocal cords, tongue, and jaw) by hand gestures, while keeping visible articulators (lips). Basically, it complements the lip-reading by various hand gestures, so that phonemes which have similar lip shapes can be differentiated. Then, considering both lip-shapes and gestures, each phoneme has a specific visual aspect. In cued speech, information is shared between two modalities: the lip modality (related to lip shape and motion) and the hand modality (related to hand configuration and hand position w.r.t the face). Figure 2.4 shows an example word from French cued speech. Cued speech is different from sign language because, among others things, cued speech addresses speech. Cued speech has the same grammar and syntax as the corresponding spoken language (English, French, ...). A deaf person learning English cued speech, for example, learns English at the same time.

Figure 2.4. An example of French cued speech: "bonjour (good morning)"

2.3. Hand Gestures in Human-Computer Interaction

Gestures, in the context of visual interpretation, can be defined as stochastic processes in the gesture model parameter space over a suitably defined time interval [17]. The stochastic property suggests the natural variability of gestures. The time interval suggests their dynamic character.

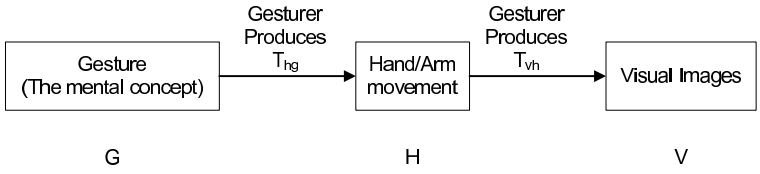


Figure 2.5. Gesture production and preception [17]

A diagram for the production and perception of gestures can be seen in Figure 2.5. The gesturer produces the gesture through hand and arm movements. The observer perceives the gestures as visual images. This model can be formulated as follows:

$$H = T_{hg}G$$

$$V = T_{vh}H$$

$$V = T_{vh}(T_{hg}G) = T_{vg}G$$

where T_{hg} is a model of hand or arm motion given gesture G, T_{vh} is a model of visual images given hand or arm motion H, and T_{vg} gives how visual images are formed given gesture G. Using a suitable model T_{vg}, gestures can be inferred from visual images

$$\hat{G} = T_{vg}^{-1}V$$

In order to build a suitable model, a detailed interpretation of human hand gestures is necessary, since different models and approaches are needed for different types of gestures. Independent from the gesture type (Figure 2.1), there are two components of a hand gesture: spatial component (hand posture) and temporal component (hand

motion). The spatial component is identified by the joint angles and the position of the fingers and the palm. The temporal component is identified by the trajectory of the hand in time. The values that identify the hand gesture must be extracted from the given image sequence for constituting a hand gesture recognition system. The needed accuracy of the extracted values differ from gesture to gesture. For some gestures, the joint angles for all the fingers may be needed with high accuracy whereas for some other gestures only the motion of the hand is important whatever the hand posture is. The exact direction of the hand motion may not be important for some gestures, but it should be analyzed for many of the manipulative gestures. Moreover, for deictic gestures, the hand posture is always the same; there is generally no hand motion, but what is important is the direction of the pointing finger.

3. State of The Art On Hand Gesture and Sign Language Recognition

The problem of sign language analysis and recognition (SLR) can be defined as the analysis of all components that form the language and the comprehension of a single sign or a whole sequence of sign language communication. The ultimate aim in SLR is to reach a large-vocabulary recognition system which would ease the communication of the hearing impaired people with other people or with computers. The components of a sign contain manual signals (MS) such as hand shape, position and movement, which form the basic components of sign languages, and non-manual signals (NMS), such as facial expressions, head motion and body posture. An SLR system requires the following components:

- Hand and body parts (face, shoulders, arms ...) detection, segmentation, and tracking.
- Analysis of manual signals
- Analysis of non-manual signals
- Classification of isolated and continuous signs

Figure 3.1 shows the general diagram of a SLR system.

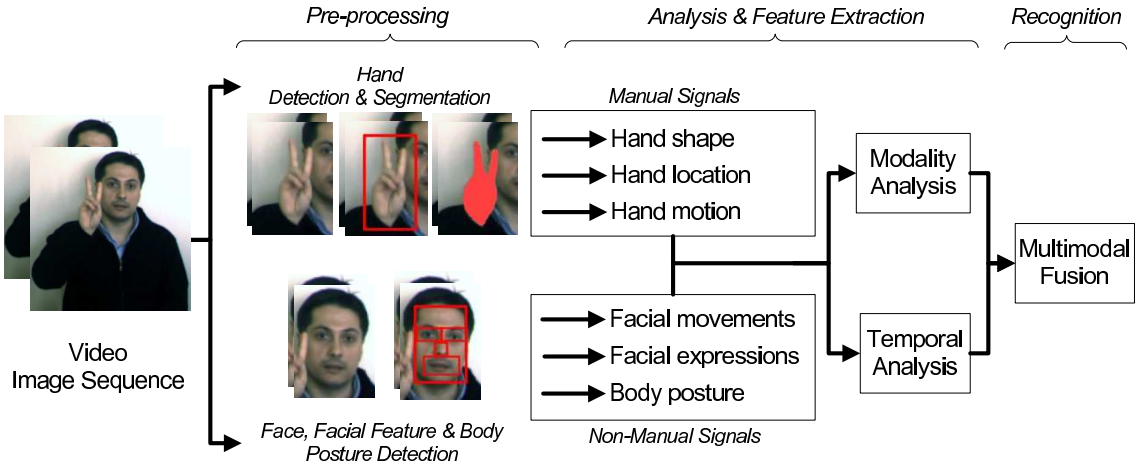


Figure 3.1. Sign language recognition system

Ultimately, an ideal system that contains these components would result in a system that takes as input a sign language video and outputs a text of sign language sentences. SLR systems can be used in many application areas such as Human-Computer Interaction on personal computers, public interfaces such as kiosks, or translation and dialog systems for human-to-human communication. SLR systems, in connection with sign synthesis, can be used in transferring sign data, where the sign is captured at one end and the output of the SLR system can be sent to the other end, where it is synthesized and displayed by an avatar. This would require a very low bandwidth when compared to sending the original sign video [24, 25, 26]. SLR systems or sign synthesis systems can also assist sign language education [2, 27, 28, 29, 30].

3.1. Detection, Tracking and Segmentation

Frontier research on hand gesture recognition and on SLR has mainly used instrumented gloves, which provide accurate data for hand position and finger configuration. These systems require users to wear cumbersome devices on their hands. However, humans would prefer systems that operate in their natural environment. Since the mid 90's, improvements in camera hardware have enabled real-time vision-based hand gesture recognition [15]. Instead of using instrumented gloves, vision based-systems, which only require one or more cameras connected to the computer, have been adopted. For a brief overview of sign language capturing techniques, interested readers may refer to [31].

Vision based systems for gestural interfaces provide a natural environment in contrast to the cumbersome instrumented gloves with several sensors and trackers that provide accurate data for motion capture. However, vision based capture methodology introduces its own challenges, such as the accurate detection and segmentation of the face and body parts, hand and finger configuration, or handling occlusion. Many of these challenges can be overcome, by restricting the environment and clothing or by using several markers such as differently colored gloves on each hand or colored markers on each finger and body part, at the expense of naturality.

Signing takes place in 3D and around the upper body region. A vision based system that uses a single camera will only capture the 2D information. However, these systems are preferred as they are simple, easy to use, and portable. The camera setup in these kind of systems is simple and the only criterion is that the camera field of view must contain the entire region of interest, which is the upper body. If the 3D information is to be captured, a stereo camera or multiple cameras can be used. In a stereo camera setting, both cameras must contain the upper body of the signer. An alternative configuration can be using two cameras, one in front, and the other on the right/left side of the signer. Additional cameras can be used to focus on the face to capture NMS in high resolution.

The main advantage of using a marker is that it makes tracking easier and helps to resolve occlusions. In a markerless environment, hand tracking and segmentation presents a bigger challenge. In sign languages, as the main part of the message is conveyed through the hands and the face, the accurate detection and segmentation of hands and face in a vision based system is a very important issue. The signing takes place around the upper body and very frequently near or in front of the face. Moreover, the two hands are frequently in contact and often occlude each other. Another problem is to decide which of these two hand regions correspond to the right and left hands. Thus, the tracking and segmentation algorithm should be accurate enough to resolve these conditions and provide the two segmented hands.

Figure 3.2. An example ASL sign, "door": Markerless hand tracking is challenging since the hands and face are in occlusion

Hand detection and segmentation can be done with or without markers. Several markers are used in the literature such as single colored gloves on each hand, or gloves

Table 3.1. Cues for hand detection and segmentation

Type of Information	**Problems**	**Assumptions/Restrictions**
Color cue	Existence of other skin colored regions	Long-sleeved clothing
	Contact of two hands	Excluding the face
	Identifying the left and right hands	Only single hand usage
Motion cue	Motion of objects other than the hands	Stationary background
	Fast and highly variable motion of the hand	Hand moves with constant velocity
Shape cue	High degree of freedom of the hand	Restricting the hand shapes

with different colors on each finger or joint. With or without a marker, descriptors of color, motion and shape information, separately or together, can be used to detect hands in images. However, each source of information has its shortcomings and restrictions. Table 3.1 lists the shortcomings of each source of information and the assumptions or restrictions on the systems that use them. Systems that combine several cues for hand segmentation have fewer restrictions and are more robust to changes in the environment [32, 33, 34].

Color information is used with the strong assumption that hands are the only skin regions in the camera view. Thus, users have to wear long-sleeved clothing to cover other skin regions such as arms [33, 35]. Face detection can be applied to exclude the face from the image sequence, leaving the hands as the only skin regions. However, this approach ignores situations where the hand is in front of the face: a common and possible situation in sign languages (see Figure 3.2). When there are two skin regions resulting from the two hands of the signer, the two biggest skin-colored regions can be selected as the two hands. This approach will fail when the two hands are in contact, forming a single skin-colored region. Another problem is to decide which of these two regions corresponds to the right and left hands and vice versa. In some studies, it is assumed that users always use or at least start with their left hand on the left and right hand on the right. Starting with this assumption, an appropriate tracking algorithm can be used to track each region. However, when the tracking algorithm fails, the users need to re-initialize the system. Some of these problems can be solved by using motion

and shape information.

Motion information can be highly informative when the hand is the only moving object in the image sequence [36]. This assumption can be relaxed by combining the motion cue with the color cue and assuming that the hand is the only moving object among the skin colored regions.

The main disadvantage of using the shape information alone comes from the fact that the hand is a non-rigid object with a very high degree of freedom. If the number of hand shapes are limited, shape information can be robustly used for detection and segmentation, even during occlusion with the face or with the other hand [37]. Thus, to achieve high classification accuracy of the hand shape, either the training set must contain all configurations that the hand may have in a sign language video, or the features must be invariant to rotation, translation, scale and deformation in 3D [38].

Once the hands are detected, this information can be used to track the hands in time. Kalman filter and particle filter based methods are state-of-the art methods used for tracking the signers' hands.

Kalman filtering [39] is one of the well known and widely used method for object tracking with its ease of use and real-time operation capability. Kalman filter assumes that the tracked object moves based on a linear dynamic system with Gaussian noise. For non-linear systems, methods based on Kalman filter are proposed, such as Extended Kalman Filter, and Unscented Kalman Filter. For hand tracking purposes, Kalman filter based methods are used with the assumption of limited and restricted motion of the hands. In [40] it is assumed that the closest region to the Kalman filter prediction that matches the color histogram of the hand is assumed to be the tracked hand, and the assignment of left and right hands are done following this assumption. Although the method is able to handle situations when the hands are in contact or in front of the face up to a degree, the restrictions for the hand movement are not valid in a natural signing environment.

The Particle Filter (PF) is another well-known example and is well suited to applications where the dynamics of the tracked object are not well-defined, or with non-linear dynamics and non-Gaussian noise conditions. Conditional density propagation (Condensation) algorithm [41] is a simple implementation of the PF and is proposed for the object tracking problem. The main idea is to make multiple hypotheses for the object at a given time and to estimate the object position by a combination of these hypotheses. The hypotheses are weighted and a highly weighted hypothesis indicates a possible position for the object. Although a PF is capable of multiple object tracking, it can not properly handle cases when the objects touch or occlude.

Both the Kalman filter and the particle filter methods need a dynamic model for the hand motion which is not easy to estimate. Although the estimations of the particle filter in the case of hand tracking are more accurate than the Kalman filter, the main disadvantage is its computational cost, which prevents its usage in real time systems. Apart from these methods, several dynamic programming approaches have also been proposed [42].

Besides the hands, several body parts such as the face, shoulders and arms should be detected [32, 43] to extract the relative position of the hands with respect to the body. This position information is utilized in the analysis of MS. Moreover, the position and motion of the face, facial features and the whole body is important for the analysis of NMS.

There are also some works that do not apply any tracking and use the output of the detection step [44]. After the detection, a foreground image is formed which contains only the hands and the face of the signer, omitting the unrelated, background pixels. The feature extraction is done on this foreground image.

3.2. Modality Processing and Feature Extraction

The modalities involved in gestured languages can be discussed from several points of view:

- The part of the body that is involved: Hands, head, facial features, shoulders, general standing. For example, sign languages use the whole upper body, hands, head, facial features, and body/shoulder motion.
- Whether the modality conveys the main message or a paralinguistic message: In sign languages, paralinguistic elements can be added to the phrase via the non-manual elements or variations of the manual elements. The main message is contained jointly in the manual (hand motion, shape, orientation and position) and non-manual (facial features, head and body motion) modalities where the non-manual elements are mainly used to complement, support or negate the manual meaning.
- Whether the modality has a meaning by itself or not: In sign languages, the manual component has a meaning by itself for most of the signs. However, this meaning can be altered by the use of the non-manual component, which is needed for full comprehension. The non-manual component alone may have a meaning by itself, especially to indicate negations.

In this section, we present analysis and classification methods for each of the modalities independently. The synchronization, correlation, and the fusion of modalities are discussed in the next sections.

3.2.1. Manual Signals

Manual signals are the basic components that form sign languages. These include hand shapes, hand motion, and hand position with respect to body parts. Manual sign language communication can be considered as a subset of gestural communication where the former is highly structured and restricted. Thus, analysis of manual signs is highly connected to hand gesture analysis [16, 17] but needs customized methods to solve several issues such as analysis of a large-vocabulary system, correlation analysis of signals and to deal with its structured nature.

Some of the studies in SLR literature concentrate only on recognizing static hand shapes. These hand shapes are generally selected from the finger alphabet or from

Table 3.2. Feature extraction for manual signs in vision based systems

Hand shape	**Hand motion**	**Hand position wrt body**
• Segmented hand • Binary hand – Width, height, area, angle [2, 45] – Log Polar Histograms [38] – Image moments [46] • Hand contour – Curvature Scale Space [47] – Active Contours [33] • 3D hand Models	• Center of mass coordinates & velocity of hands [2] • Pixel motion trajectory [48] • Discrete definitions of hand motion and relative motion of two hands [38]	• Distance to face [2] • Distance to body parts [48] • Discrete body region features [38, 49]

static signs of the language [50, 51, 52]. However, a majority of the signs in many sign languages contain significant amount of hand motion and a recognition system that focuses only on the static aspects of the signs has a limited vocabulary. Hence, for recognizing hand gestures and signs, one must use methods that are successful on modeling the inherent temporal aspect of the data.

Grammatical structures in the language are often expressed as systematic variations of the base manual signs. These variations can be in the form of speed, tension, and rate [53, 54]. Most of the SLR systems in the literature ignore these variations. However, special care must be paid to variations, especially for continuous signing and in sign-to-text systems.

Table 3.2 summarizes the features used in the literature for each of the modalities that constitute a manual signal. The details are given in the following sections.

3.2.1.1. Hand Shape. Hand shape is one of the main modalities of the gestured languages. Apart from sign language, hand shape modality is widely used in gesture controlled computer systems where predefined hand shapes are used to give specific

commands to the operating system or a program. Analysis of the hand shape is a very challenging task as a result of the high degree of freedom of the hand. For systems that use limited number of simple hand shapes, such as hand gesture controlled systems (hand shapes are determined manually by the system designer), the problem is easier. However, for sign languages, the unlimited number and the complexity of the hand shapes make discrimination a very challenging task, especially with 2D vision based capture systems.

In sign languages, the number of hand shapes is much higher. For example, without considering finger spelling, American Sign Language (ASL) has around 150 hand shapes, and in British Sign Language (BSL) there are 57 hand shapes. These hand shapes can be further grouped into around 20 phonemically distinct subgroups.

For the analysis of hand shapes, a vast majority of the studies in the literature use appearance based methods. These methods extract features of a hand shape by analyzing a 2D hand image and are preferred due to their simplicity and low computation times, especially for real time applications. These features include region based descriptors (image moments [46], image eigenvectors, Zernike moments, Hu invariants [55, 56, 7], or grid descriptors) and edge based descriptors (contour representations [37], Fourier descriptors, or Curvature Scale Space (CSS) [47] descriptors). 2D deformation templates or active contours [33] can be used to find the hand contour. When the position and angles of all the joints in the hand are needed with high precision, 3D hand models should be preferred. The 3D hand model can be estimated either from a multiple camera system by applying 3D reconstruction methods, or in a single camera setting, the 2D appearances of the hand are matched with 3D hand models in the shape database [57]. However, computational complexity of these methods currently prevents their use in SLR systems.

3.2.1.2. Hand Motion. In gestured communication, it is important to determine whether the performed hand motion conveys a meaning by itself.

In sign languages, the hand motion is one of the primary modalities that form the sign, together with the hand shape and location. Depending on the sign, the characteristic of the hand trajectory can change, requiring different levels of analysis. For example, some signs are purely static and there is no need for trajectory analysis. The motion of the dynamic signs can be examined as either of two types:

1. Signs with global hand motion: In these signs, the hand center of mass translates in the signing space.
2. Signs with local hand motion: This includes signs where the hand rotates with no significant translation of the center of mass, or where the finger configuration of the hand changes.

For signs with global hand motion, trajectory analysis is needed. For signs with local motion, the change of the hand shape over time should be analyzed in detail, since even small changes of the hand shape convey information.

The first step of hand trajectory analysis is tracking the center of mass of each segmented hand. Hand trajectories are generally noisy due to segmentation errors resulting from bad illumination or occlusion. Thus, a filtering and tracking algorithm is needed to smooth the trajectories and to estimate the hand location when necessary. Moreover, since hand detection is a costly operation, hand detection and segmentation can be applied not in every frame but less frequently, provided that a reliable estimation algorithm exists. For this purpose, algorithms such as Kalman filters and particle filters can be used, as explained in Section 3.1, and the estimations of these filters for the position, and its first and second order derivatives, the velocity and the acceleration of the hand are used as hand motion features. Based on the context and the sign, hand coordinates can be normalized with respect to the reference point of the sign, as discussed in Section 3.2.1.3.

Relative motion and position of each hand with respect to the other is another important feature in sign language. The synchronization characteristics of the two hands differ from sign to sign, i.e., the two hands can move in total synchronization;

one hand can be stationary and the other can be moving; they can be approaching or moving away.

Although the hand motion features are continuous values, they can be discretized for use in simpler models, especially when there is low amount of training data. In [46], the authors define a linguistic feature vector for the sign language. Their feature vector includes the discretized values for the position of the hands relative to each other, position of hands relative to key body locations, relative movement of the hands and the shape of the hands (the class index).

3.2.1.3. Hand Position. The location of the hand must be analyzed with respect to the context. It is important to determine the reference point on the space and on the hand. In sign languages, where both the relative location and the global motion of the hand are important (see Figure 3.3), the continuous coordinates and the location of the hand with respect to body parts should be analyzed. This analysis can be done by using the center of mass of the hand. On the other hand, for pointing signs, using center of mass is not appropriate and the coordinates of the pointing finger and the pointing direction should be used. Since signs are generally performed in 3D space, location analysis should be done in 3D if possible. Stereo cameras can be used to reconstruct 3D coordinates in vision based systems.

3.2.2. Non-manual Signals

Non-manual signals are used in sign language either to strengthen or weaken or sometimes to completely change the meaning of the manual sign [58, 59]. These include facial expressions, facial movements, and body posture. For example, by using the same MS but different NMS, the ASL sign HERE may mean NOT HERE, HERE (affirmative) or IS HERE. The non-manual signs can also be used by themselves, especially for negation [58, 59]. As opposed to studies that try to improve SLR performance by adding lip reading to the system [60], analysis of NMS is a must for building a complete SLR system: two signs with exactly the same manual component

(a)

(b)

(c)

Figure 3.3. Possible reference points on the signing space. Examples from ASL. (a) CLEAN sign: Hand location w.r.t the other hand, (b)DRINK sign: Hand location w.r.t the mouth, (c) HERE sign: Hand location w.r.t the body

can have completely different meanings. Some limited studies on non-manual signs attempt to recognize only the NMS without the MS. In [61, 62], head movements and in [63], facial expressions in ASL are analyzed. In SLR literature, there are only a few studies that integrate manual and non-manual signs [3].

The synchronization characteristics of non-manual signals and manual signs should be further analyzed. In a continuous sign language sentence the non-manual signal does not always coincide with the manual sign. It may start and finish before or after the manual sign. It may last through multiple manual signs or the whole sentence. Thus, the joint analysis of the manual signs and non-manual signals requires the fusion of modalities in different time scales.

3.3. Recognition

Signs comprise dynamical elements. A recognition system that focuses only on the static aspects of the signs has a limited vocabulary. Hence, for recognizing hand gestures and signs, one must use methods that are capable of modeling inherent temporal characteristics. Researchers have used several methods such as neural networks, HMMs, Dynamic Bayesian Networks (DBN), Finite State Machines (FSM) or template matching [16]. Among these methods, HMMs have been used the most extensively and have proven successful in several kinds of SLR systems. Initial studies on vision-based SLR started around mid 90's and focused on limited vocabulary systems, which could recognize 40-50 signs. These systems were capable of recognizing isolated signs and also continuous sentences, but with constrained sentence structure [64]. In 1997, in [65], an HMM structure was applied to recognize a 53 sign vocabulary. The scalability problem is addressed in later studies, where an approach based on identifying phonemes/components of the signs [66] rather than the whole sign has been adopted. The advantage of identifying components is the decrease in the number of subunits that should be trained, which in turn will be used to constitute all the signs in the vocabulary. With component-based systems, large vocabulary SLR can be achieved to a certain degree. For a list of SLR systems in the literature, users may refer to the excellent survey in [58].

The merit of a recognition system is its generalization power of a learned concept to new instances. In sign language recognition, there are two kinds of new instances for any given sign: (1) signing of a signer whose other signing videos are put in the training set; (2) signing of a new signer. The former is called as the signer dependent and the latter as the signer independent experiments. The real performance of a sign language recognition system must be measured in a signer-independent experiment. Most of the studies in the literature are signer dependent and lack an analysis of signer independent performance of their system. Experiments on signer independent systems show that the recognition accuracy of such systems are around at least 10% lower than the signer dependent ones [2, 67]. There are a few studies that investigates signer adaptation techniques [53].

In the next sections, we present a detailed state of the art for sign based approaches and sub-unit based approaches for both isolated and continuous signing tasks.

3.3.1. Sign Based Approaches

Sign based approaches use the sign as the smallest unit and model each sign with a different temporal model. As in speech recognition, HMMs have become state-of-the-art for sign modeling and most of the recent works use HMMs or variants as the temporal sign model.

3.3.1.1. Recognition of Isolated Signs. Isolated sign recognition deals with the recognition of signs that are performed alone, without any signs before or after the performed sign. Thus, the sign is performed, without being affected by the preceding or succeeding signs. The importance of the research on isolated sign recognition is that it enables the finding of better mathematical models and features that represent the performed sign.

In HMM-based approaches, the temporal information of each sign is modeled by a different HMM. For a test sign, the model that gives the highest likelihood is selected as the best model and the test sign is classified as the sign of that model. One of the main challenges is the integration of the two hands and the different features for each hand (shape, motion, position and orientation) into the HMM model [68, 69].

In Tables 3.3 and 3.4, we list several selected SLR systems in the literature. The tables show that most of the recent systems use HMMs or HMM variants for classification. Although the recognition rates of device or vision based systems are comparable, the shortcoming of vision based systems is that they make a lot of assumptions and introduce several restrictions on the capturing environment.

Computational problems arise when the vocabulary size is increased. Using a different model for each sign, trying all models and selecting the best requires a lot of

Table 3.3. SLR systems for isolated signs that use a specialized capturing device

Work	Sign Dataset	Classification Method	Accuracy %*
[72]	102 CSL	Boosted HMMs	92.7 SD
[70]	5113 CSL	Fuzzy Decision Tree, Self Organizing Feature Maps, HMM	91.6 SD 83.7 SI
[71]	95 Auslan	Instance based learning, decision trees	80 SD 15 SI
[76]	10 JSL	Recurrent NN	96 SD

**SD: Signer Dependent, SI: Signer Independent*

computation and prevents the use of such systems in large vocabulary tasks.

In [70], the authors present a hierarchical structure based on decision trees in order to be able to expand the vocabulary. The aim of this hierarchical structure is to decrease the number of models to be searched, which will enable the expansion of the vocabulary since the computational complexity is relatively low. They used a sensored glove and a magnetic tracker to capture the signs and achieved 83% recognition accuracy, at less than half a second average recognition time per sign, in a vocabulary of 5113 signs. Decision trees are also used in [71] together with instance based learning. An adaptive boosting (AdaBoost) strategy to continuous HMMs is presented in [72].

The pronunciation differences between signer is handled with automatic clustering techniques in [73]. They use the tangent distance within the Gaussian densities of the continuous HMM. They achieve an accuracy of 78.5% with the automatic clustering and the tangent distance. A hybrid structure that uses HMMs and Auto-regressive HMMs is presented in [45]. In [46], a discrete feature vector representing the linguistic properties of the signs is presented. The classification is based on Markov chains and Independent Component Analysis (ICA) and an accuracy of 97.67% is achieved on a BSL dataset. In [74], a vision based system for Arabic sign language (ArbSL) is presented. Curvature scale space is used as hand shape features in [47]. In [75], a colored glove with different colors on each finger is used. With this information, hand shape features such as the angles between fingers were easy to extract. An HMM structure is used where the number of states and the number of mixture components are defined dynamically.

Table 3.4. Vision based SLR systems for isolated signs

Who	Sign Dataset	Capture Restrictions	Hand Features Shape	Motion	Position	Classification Method	Accuracy %*
[2]	19 ASL, with NMS	colored gloves	2D app. based	Position, velocity	Distance to face	HMM based	99 SI (MS), 85 SI (MS+NMS)
[73]	50 ASL, with PD**	Static bg.	2D app. based features of whole image			HMM with tangent distance	78.5 SD
[45]	439 CSL	colored gloves	2D app. based		Distances of hands to body regions & each other	HMM, Autoregressive HMM	96.6 SD
[74]	50 ArbSL	colored gloves	2D binary hand		Hand coords. wrt face	HMM	98 SD
[46]	43 BSL	colored gloves	Classified hand shape	Movement type	Positions of hands wrt body regions & each other	Markov Chains, ICA	97.67 SD
[47]	20 TwSL, single handed	Dark, static bg.; dark long-sleeved clothes	2D CSS	-	-	HMM	98,6 SD
[75]	262 NethSL	Uniform bg.; stationary head; dark long-sleeved clothes	2D Moments	-	-	HMM	91.3 SD

SD: Signer Dependent, SI: Signer Independent

***PD: Pronunciation differences*

3.3.1.2. Recognition of Continuous Signs. Recognizing unconstrained continuous sign sentences is another challenging problem in SLR. During continuous signing, signs can be affected by the preceding or succeeding signs. This effect is similar to co-articulation in speech. Additional movements or shapes may occur during transition between signs. These movements are called movement epenthesis [77]. These effects complicate the explicit or implicit segmentation of the signs during continuous signing. To solve this problem, the movements during transitions can be modeled explicitly and used as a transition model between the sign models [78, 33, 68]. In Tables 3.6 and 3.5, we list several selected SLR systems proposed for continuous signing, with vision based and capturing device based systems respectively.

One of the first studies on continuous sign language recognition is presented in [64]. The authors propose a vision based system and experimented with both colored glove based and skin color based tracking. Their system uses HMMs and a grammar to recognize continuous signing. With colored glove based tracking, they achieve a sign-level recognition accuracy of 99%. Sentence level accuracies are not reported.

In [65], the authors address the co-articulation effects and model the transition movements between signs, which are called the movement epenthesis. These transitions are identified automatically by applying a k-means clustering technique on the start and end points of the signs. They constructed an epenthesis model recognition technique in which each sign is followed by a transition cluster. On a 53 sign vocabulary and 489 sentences, they compare context dependent and independent approaches with or without the epenthesis modeling. They also compare unigram or bigram models for sign recognition. The best accuracy, 95.8%, is obtained with epenthesis modeling with bigrams, in comparison to an accuracy of 91.7% which is obtained by context dependent bigrams without epenthesis modeling.

A signer-independent continuous Chineese Sign Language (CSL) recognition with a divide-and-conquer approach is presented in [67]. The authors use a combination of a Simple Recurrent Network (SRN) and HMMs, SRN is used to divide the continuous signs into the subproblems of isolated CSL recognition and the outputs of SRN are used as the states of an HMM. The accuracy of the signer independent tests is 7% lower than the signer dependent one.

An automatic vision based Australian sign language (Auslan) recognition system is presented in [33]. The authors model each sign with HMMs, with features that represent the relative geometrical positioning and shapes of the hands and their directions of motion. Their system achieved 97% recognition rate on sentence level and 99% success rate at a word level, on 163 test sign phrases, from 14 different sentences.

In [78], a methodology based on Transition-Movement models (TMMs) for large-vocabulary continuous sign language recognition is proposed. TMMs are used to handle the transitions between two adjacent signs in continuous signing. The transitions are dynamically clustered and segmented and these extracted parts are used to train the TMMs. The continuous signing is modeled with a sign model followed by a TMM. The recognition is based on a Viterbi search, with a language model, trained sign models and TMM. The large vocabulary sign data of 5113 signs is collected with a sensored glove and a magnetic tracker with 3000 test samples from 750 different sentences. Their

Table 3.5. SLR systems for continuous signs that use a specialized capturing device

Work	Sign Dataset	Classification Method	Accuracy %*
[78]	5113 CSL signs, 750 sentences**	Transition Movement Models (TMM)	91.9 sign-level SD
[67]	208 CSL signs, 100 sentences	HMM, Self Organizing Map, Recurrent NN	92.1 SD 85 SI
[65]	53 ASL signs, 486 sentences	Movement epenthesis models	95.8 sign-level SD

**SD: Signer Dependent, SI: Signer Independent*

***Total number of different sentences*

Table 3.6. Vision based SLR systems for continuous signs

Who	Sign Dataset	Capture Restrictions	Shape	Hand Features Motion	Hand Features Position	Classification Method	Accuracy %*
[33]	21 Auslan signs, 14 sentences**	Dark, static bg.; dark long-sleeved clothes	2D geometry based	Movement direction	Geometric features wrt face	HMM	99 sign-level, 97 sentence- level (SD)
[64]	40 ASL, 494 sentences**	Static bg.; dark long-sleeved clothes; colored gloves	2D Moments	-	-	HMM	99 sign-level*** (SD)

**SD: Signer Dependent, SI: Signer Independent*

*** Total number of different sentences*

**** Sentence level accuracy is not reported*

system has an average accuracy of 91.9%.

3.3.2. Sub-unit Based Approaches

As the vocabulary size increases, computational complexity and scalability problems arise. One of the solutions to this problem is to identify phonemes/subunits of the signs like the phonemes of speech. The advantage of identifying phonemes is to decrease the number of units that should be trained. The number of subunits is expected to be much lower than the number of signs. Then, there will be a smaller group of subunits that can be used to form all the words in the vocabulary. This will enable large-vocabulary sign language recognition with tractable computation times.

However, the phonemes of sign language are not clearly defined. Some studies use the number of different hand shapes, motion types, orientation, or body location as the phonemes [79, 68]. Others try to automatically define the phonemes by using clustering techniques [80]. Table 3.7 lists selected SLR systems that use subunit based approaches.

One of the first studies that use subunits instead of whole signs is the work of Vogler and Metaxas [66]. The subunits of the signs are identified implicitly by looking at the geometric properties of the motion trajectory, such as hold, line, or plane. Their system achieved a 89.9% sign accuracy on a database of 53 signs and 486 sentences in a continuous signing task.

In their later study [81], they define a new model based on a sequential phonological model of ASL, the Movement and Hold model, which states that the ASL signs can be broken into movements and holds. These are considered as phonemes of the signs and they are explictly defined in the movement-hold model. In addition, they model the transitions between the signs with the movement epenthesis models form their earlier work [65]. For each phoneme an HMM is trained and the combination of these phoneme models form a sign with the epenthesis models to define the transitions between the signs in continuous signing. Experiments with a 22 word vocabulary, modeled with 89 subunits (43 phonemes, 46 epenthesis models), yield similar word-level recognition results with word and phoneme-based approaches, 92.95%, 93.27% respectively.

In a recent work [68], they modeled the movement and shape information of the signs in separate HMM channels and used Parallel HMMs for this task. They model the right and left hand information in separate channels as well, as presented in their earlier work [69]. Their system achieved 87.88% sign-level and 95.51% sentence-level accuracy.

In [80], an automatic subunit extraction methodology for CSL is presented. They use HMMs to model each sign and assume that each state in the HMM represents a segment. The subunits are extracted from these segments by using a temporal clustering algorithm. Their system achieved a 90.5% recognition rate

In [79], the authors define *etymon* as the smallest unit in a sign language. The etyma are not automatically extracted and defined explicitly. 2439 etyma are defined for CSL and HMMs are trained for each etymon. They made a comparison between

Table 3.7. SLR systems with subunit based approaches

Who	Dataset	Phonemes/ Subunits	Capture Method	Subunit Determination Method	Classification Method	Accuracy %*
[79]	5100 CSL	2439 etyma	Device based	Manual	HMM	96 sign based (SD), 93 etymon based
[80]	5113 CSL	238 subunits	Device based	Automatic, temporal clustering	HMM	90.5 sign-level (SD)
[68]	22 ASL signs, 499 sentences	3 channels	Device based	Manual, movement and shape channels	Parallel HMM	95.5 sign-level (SD), 87.9 sentence-level
[81]	22 ASL signs, 499 sentences	89 subunits	Device based	Manual, Movement-Hold model	HMM	92.9 sign based (SD), 93.3 phoneme-based
[66]	53 ASL signs, 499 sentences	3 subunits	Device based	Automatic, geometric movement properties	HMM	89.9 sign-level (SD)

**SD: Signer Dependent, SI: Signer Independent*

etyma-based and sign-based approaches on a 5100 sign CSL database, collected with a specialized capturing device. However, their analysis shows that the sign-based method has both better accuracy and lower recognition times.

3.4. Databases

One of the problems of the research on sign language recognition is the nonexistence of benchmark databases. Many techniques are developed by independent researchers but these works lack a structured evaluation on a benchmark dataset and comparison with the other proposed techniques in the literature.

Sections 3.4.1,3.4.2, and 3.4.3 present the databases used in this study. Section 3.4.4 summarizes some of the publicly available sign language databases.

3.4.1. IDIAP Two Handed Gesture Database

IDIAP gesture dataset is a small gesture dataset, with seven two-handed gestures to manipulate 3D objects [82]. The gestures are a push gesture and rotate gestures in six directions: back, front, left, right, down, up. Two cameras are used, positioned on the left and right of the user. The users wear gloves: a blue glove on the left and a yellow glove on the right hand. The training set contains 280 examples recorded from four people and the test set contains 210 examples recorded from three different

people. More information on the database can be found in [82]. The experiments in Chapter 6 is performed on this database.

3.4.2. eNTERFACE'06 Sign Language Database

In the eNTERFACE06 ASL Database, there are eight base signs that represent words and a total of 19 variants which include the systematic variations of the base signs in the form of non-manual signs, or inflections in the signing of the same manual sign. Some signs are differentiated only by the head motion (the non-manual component); some only by hand motion (the manual component) and some by both. A single web camera is used for the recordings with 640x480 resolution and 25 frames per second. The camera is placed in front of the subject. The database is collected from eight subjects and each subject performs five repetitions of each sign [2], while wearing a marker, a colored glove. The experiments in Chapters 5 and 6 are performed on this database.

3.4.3. TRT Signed Turkish Database

This database is recorded from the Turkish radio television's broadcast news for the hearing impaired. The news video consists of three major information sources: sliding text, speech and signs. The three sources in the video convey the same information via different modalities. The news presenter signs the words as she talks. The signing in these news videos is considered as signed Turkish. The data contains 132 recordings, where each recording is around 10 minutes. There are two different signers in the videos. The resolution of the videos are 384 x 288 with 25 frames per second. The experiments in Chapter 4 is performed on this database.

3.4.4. Other Sign Language Databases

A few other sign language databases, some of which are collected for non-recognition purposes, are publicly available in the internet. Here we list some of them: British Sign Language Database[46], Purdue American Sign Language Database [83], American Sign

Language lexicon video dataset [84].

3.5. Discussion

Isolated SLR achieved much attention in the past decade and systems were proposed that have high accuracies in the reported databases of a wide range of sign languages from all over the world. However these datasets contain different range and number of signs that may be recorded with strong restrictions (slow speed, non-native signers, unnatural signing ...). There are no benchmark sign datasets that researchers can test and to which they can compare their systems. Although there are some publicly available datasets (Section 3.4.4), these datasets have not yet become benchmark datasets of SLR researchers.

Current challenges of SLR can be summarized as continuous signing, large vocabulary recognition, analysis and integration of non-manual signals, and grammatical processes in manual signing. Although these aspects are mentioned by several researchers of the field [85, 58], the amount of research in these areas are still limited. Significant progress can be made by close interaction of SLR researchers with sign language linguists.

4. Marker Free Hands and Face Tracking

4.1. Introduction

Tracking the hand and face of a signer during unconstrained, natural signing is a difficult problem. Many of the proposed methods for hands and face tracking have restrictive assumptions like placing markers on the hands or restricting the movements so that the hands do not touch or occlude each other or the face. However, in sign languages, the contact of hands and the face is very frequent and purposeful; as it is a building block for many signs. Although using markers solves most of the tracking, identification and segmentation problems, this is an unrealistic assumption in many applications. The application that we focus on, is the joint tracking of the face and the hands in the videos of a news reporter in news for the hearing impaired. The news reporter performs the corresponding signs with her hands as she speaks. The videos involve many difficulties such as unconstrained clothing, changing speakers, changing backgrounds, frequent contacts and occlusions between the hands and the face, low resolution, and motion blur. Therefore, a robust tracking algorithm that is capable of tracking the hands and the face, without any markers, is needed.

We pose this problem as a multiple identical object tracking problem with dependencies between the objects. The objects, for our case the hands and the face, are identical in terms of the skin color and there are strong correlations between their movements, especially during interactions. These interactions usually result in the occlusion of one object by the other: The hand occludes the face and points to a facial feature. Two hands often come in contact, and they may cross each other. The tracking algorithm should be capable of tracking and correctly labeling each accurately even in these difficult cases.

For tracking multiple identical objects in a scene, the classical approach is to first perform a detection and to associate the detected candidates with the tracked objects afterwards. Most of these methods are based on multiple hypothesis tracking

[86], which is a well known data association algorithm. The data association is either done immediately by looking at the current and past observations [40], or delayed for a number of frames or until the end of the sequence [42, 87].

Probabilistic tracking algorithms [88] make multiple hypotheses for the object at a given time and estimate the object position by a combination of these hypotheses. The Particle Filter (PF) is a well-known example and is well suited to applications where the dynamics of the tracked object are not well-defined. Conditional density propagation algorithm [41] is a simple implementation of the PF and is proposed for the object tracking problem. Although a PF is capable of multiple object tracking, it can not properly handle cases when the objects touch or occlude. To solve this problem several PF based methods are proposed that can track multiple objects. BraMBle [89] is a joint PF approach for tracking humans, where all the humans in the scene are considered as a single particle and tracked jointly. In [90], an approach for joint detection and tracking of variable number of colored objects is presented. In [91], independent PFs are used to track variable number of humans. The interactions between the humans are handled by competitive condensation, which breaks the independence of the filters. Each PF updates the weights of the particles of other filters with respect to their proximity to its current estimate. Khan et al. [92] aim to track a variable number of ants and they model the interactions between the ants as a Markov Random Field (MRF). They use joint PF when there are interactions and independent PFs otherwise. The motivation behind this approach is that tracking with joint PFs is not efficient as the complexity increases exponentially with the number of objects to track.

In this work, we propose a PF based method that can robustly track the hands and the face during natural signing. We use a joint PF to track a maximum of three objects: two hands and the face. However, the number of objects may vary during the video: One or both hands may disappear and re-appear. The complexity of the joint PF is reduced by embedding Mean Shift (MSh) tracking [93], which allows us to achieve similar tracking accuracy by using substantially fewer particles. We handle the occlusions by updating the likelihood of the particles with respect to their proximity and forcing them to be as separate as possible. The method is robust to occlusions of

the hands and the face and is able to recover fast if the tracking fails. We evaluate the performance of our method on two different videos, seven minutes each, that are recorded from the broadcast news for the hearing impaired. We achieved a tracking accuracy of 99% for the face and about 96% for the two hands.

Section 4.2 presents our joint PF approach for hands and face tracking. In section 4.3, we present a semi-independent PF approach for comparison purposes. The tracking results on the test videos, and comparison with different PF approaches with or without mean shift are given in Section 4.4.

4.2. Joint PF for Hands and Face Tracking

The posterior distribution of a PF is intrinsically multi-modal and the local maxima of the distribution indicate possible target positions [41]. Particle clustering or independent PFs can be used to track multiple objects in a scene; however, these approaches can only be applied when the objects do not interact. When two identical objects are in contact, the distribution becomes uni-modal and the two objects are considered as one. The object assignment problem before and after the interaction is cumbersome since the object information before the interaction is lost. Hence, in the case of contact, special care must be taken in order to ensure the continuity of accurate tracking during and after the interaction. We propose to use a joint PF that calculates a combined likelihood for all objects by modeling the likelihood of each object with respect to others.

A PF attempts to approximate the posterior distribution using the sequential Bayes method. The posterior distribution gives the probability of the object state at time t given the initial state and the observations.

$$p(\mathbf{x}_t|\mathbf{x}_0, \mathbf{Z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{Z}_{1:t}) d\mathbf{x}_{t-1} \tag{4.1}$$

where $\mathbf{x}_t$ is the system state at time t and $\mathbf{x}_0$ is the initial state of the system, which

can be determined by a prior distribution. $\mathbf{Z}_{1:t}$ denotes the observations from time 1 to t:

$$\mathbf{Z}_{1:t} = \{\mathbf{z}_t, \mathbf{z}_{t-1} \cdots \mathbf{z}_1\} \tag{4.2}$$

The approximation is done by weighted particles

$$\{(\mathbf{x}_t^n, \pi_t^n) : n = 1, \ldots, N\} \tag{4.3}$$

where N is the number of particles, $\mathbf{x}_t^n$ is the state vector of the n^{th} particle at time t, and π_t^n is its weight.

A state transition function and a measurement function is needed to track an object. The state transition function, $f(.)$, determines the next state of a particle at time t from its state at time $t-1$. The measurement function, $h(.)$, calculates the similarity between the particle at time t and the tracked object. The two functions are defined as

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) \tag{4.4}$$

$$\mathbf{z}_t = h(\mathbf{x}_t) \tag{4.5}$$

where the function $f(.)$ determines the dynamics of the tracked object. The function $h(.)$ models the appearance of the object and returns a likelihood value. The accuracy of a PF depends on the modeling power of these two functions.

The state of the tracked object at time t can be estimated by the weighted averaging of the particles at time t. To obtain more accurate results, we calculate the

weighted average over particles that have weights over a threshold (Φ_p).

$$\begin{aligned} \varphi_t &= \begin{cases} 1, \ if \ \pi_t^n > \Phi_p \\ 0, \ otherwise \end{cases} \\ \hat{\mathbf{x}}_t &= \frac{\sum_{n=1}^{N} \mathbf{x}_t^n \cdot \pi_t^n \cdot \varphi_t}{\sum_{n=1}^{N} \pi_t^n \cdot \varphi_t} \end{aligned} \tag{4.6}$$

4.2.1. Object Description

The state vector for a single object consists of the position, the velocity and the shape parameters. The shape parameters are selected as the width, the height and the angle of an ellipse surrounding the object. Thus, for a single object, i, we have a seven dimensional state vector,

$$\begin{aligned} \mathbf{x}_t^{n,i} &= [P_t^n, V_t^n, S_t^n]^T \\ P_t^n &= (x_t^n, y_t^n)^T \\ V_t^n &= (v_{x_t}^n, v_{y_t}^n)^T \\ S_t^n &= (a_t^n, b_t^n, \theta_t^n)^T \end{aligned} \tag{4.7}$$

where x_t^n, and y_t^n show the center of the n^{th} particle at time t, $v_{x_t}^n$, $v_{y_t}^n$ show its velocity, and a_t^n, b_t^n, θ_t^n show the two axes of the ellipse and its angle, respectively.

Then, the joint particle is a single vector containing all the objects in the scene:

$$\mathbf{x}_t^n = \{\mathbf{x}_t^{n,f}, \mathbf{x}_t^{n,r}, \mathbf{x}_t^{n,l}\}^T \tag{4.8}$$

where f, r, l are indexes to the face, right and left hands. In the rest of the paper, we will refer to $\mathbf{x}_t^n$ as the joint particle and $\mathbf{x}_t^{n,i}$ as the particle or as the sub-particle, alternatively.

4.2.2. Dynamic Model

For each object, the position and the velocity parameters are modeled by a damped velocity model and the shape parameters are modeled by a random walk model.

$$\begin{aligned} V_t &= \lambda_v V_{t-1} + \sigma\mathcal{N}(0,1) \\ P_t &= P_{t-1} + V_t \\ S_t &= S_{t-1} + \mathcal{N}(0,1) \end{aligned} \tag{4.9}$$

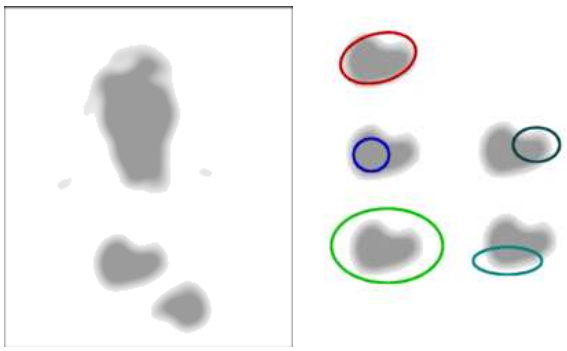
For multiple objects in the joint PF, the dynamic model in Equation 4.9 is applied to each object. We additionally apply mean shift [93] to the sub-particles of each object independently. The mean shift algorithm moves the particle centers to the areas with high skin color probability. This allows us to use particles effectively, since the particles with low weights will be less likely. As a result, a PF with mean shift needs fewer particles than a standard PF.

4.2.3. Appearance Model

We model the hands and the face by their skin color and determine the skin color pixels in the input image by a trained Gaussian mixture model [94]. The resulting grey level image is smoothed by a Gaussian kernel and using low and high threshold values, we form the thresholded skin probability image that we use for likelihood calculation (Figure 4.1(a)). This image has a positive probability for skin color pixels and zero probability for other colors.

To calculate the likelihood of a single object, we make two measurements based on the ellipse that is defined by the state vector of the particle:

- A: The ratio of the skin color pixels to the total number of pixels inside the



(a) (b)

Figure 4.1. Likelihood function definition: (a) The thresholded image that is used in likelihood calculation, (b) hand region and different particles. The likelihood function gives the highest likelihood for the particle at the top

ellipse.

- B: The ratio of the skin color pixels to the total number of pixels at the ellipse boundary.

These two ratios are considered jointly in order to make sure that our measurement function gives high likelihood to particles that contain the whole hand without containing many non-hand pixels. If we do not take the ellipse boundary into account, smaller ellipses are favored and particles tend to get smaller. We design our measurement function Equation 4.10 to return a high likelihood when A is as high as possible and B is as low as possible:

$$z_t^{n,i} = \begin{cases} 0 & , \ if \ A < \Phi_p \\ 0.5 \cdot A + 0.5 \cdot (1 - B) & , \ otherwise \end{cases} \tag{4.10}$$

where $z_t^{n,i}$ denotes the likelihood of a single object, i, for the n^{th} particle.

The first line in Equation 4.10 is required to assign low likelihood to particles that have zero or very few skin color pixels. Otherwise these particles will receive 0.5 likelihood value even if they do not contain any skin color pixels. The equation takes its highest value when there are no skin colored pixels at the boundary ($B = 0$), and when all the inner pixels are skin colored. Figure 4.1(b) shows the grey-level hand

image and possible particles. The particle at the top receives the highest likelihood by Equation 4.10 where as the other particles receive lower likelihoods.

4.2.4. Joint Likelihood Calculation

A joint particle is a combination of sub-particles which refer to the objects we want to track, i.e. two hands and the face (Equation 4.8). We calculate the joint likelihood with respect to the following:

1. The likelihood of a single object based on the appearance model ($\mathbf{z}_{1t}^n$);
2. The distance of each object to the other objects to handle interactions ($\mathbf{z}_{2t}^n$);
3. The amount of motion on the face area to handle hand over face occlusions ($\mathbf{z}_{3t}^{n,i}$);
4. Additional constraints on respective object locations ($\mathbf{z}_{4t}^n$).

We define partial likelihoods for each criterion above and calculate the joint likelihood by multiplication of the partial likelihoods:

$$\mathbf{z}_t^n = \mathbf{z}_{1t}^n \cdot \mathbf{z}_{2t}^n \cdot \mathbf{z}_{3t}^{n,i} \cdot \mathbf{z}_{4t}^n, \quad i \in \{r, l\} \tag{4.11}$$

For the first criterion, we multiply the likelihoods of each sub-particle for the joint particle n at time t. We calculate the partial likelihood for the first criterion, $\mathbf{z}_{1t}^n$, as:

$$\mathbf{z}_{1t}^n = \begin{cases} \Pi_{i \in \{f,r,l\}} z_t^{n,i} & , \; if \; \forall i \; z_t^{n,i} \neq 0 \\ \epsilon & , \; otherwise \end{cases} \tag{4.12}$$

If likelihood of any sub-particle is zero, we set the likelihood to a very small value, ϵ, instead of setting it to zero. If an object has disappeared, for all the joint particles, we will have a zero value for the sub-particles of the disappearing object. If we assign zero joint likelihood in these cases, even if the other sub-particles have very high likelihoods, because of the disappearing object, all joint particles will have zero particles.

Figure 4.2. Tracking during hand-hand occlusion. Blue and green ellipses show the estimates for the right and left hands

To handle interactions between the objects, we should assign low likelihood if the sub-particles are close to each other. This is done by a function of the distance between the particles.

$$\mathbf{z}_{2t}^{n} = \Pi_{i,j\in\{f,r,l\},\ i\neq j}(1 - exp(-\alpha \cdot ||\mathbf{x}_{t}^{n,i}, \mathbf{x}_{t}^{n,j}||)) \tag{4.13}$$

where $||.,.||$ denotes the distance and $\alpha > 0$ is a parameter that controls the effect of the distance on the likelihood.

Equation 4.13 prevents sub-particles from coincidence. For two hands in occlusion, its effect is having two ellipses next to each other, covering a part of the combined target, but never at the same position (See Figure 4.2). Similarly for the hand-face occlusion, it prevents the hand from being at the same position with the face center, or being too close. However, since the face size is bigger than the hand and since the objects are identical in terms of their skin color, Equation 4.13 gives a probability not low enough to cancel out a hand sub-particle which is mistakenly located somewhere in the face region. We solve this problem by considering the amount of motion in the face area. Since the signer's head is relatively stationary, it allows us to make the assumption that the head movement in two consecutive frames is relatively small. Using this assumption, the difference of the face regions of two consecutive frames gives us the movement of face pixels. When the hand enters the face region, the total movement will be high and when it is far away, the total movement will be small. Using this information, we only allow a hand sub-particle to get close to the face if there is

(a) (b)

Figure 4.3. Tracking during hand-face occlusion: (a) Tracking result when the hand is in front of the face. (b) The movement of the face pixels

considerable amount of movement in the face area.

$$\mathbf{z}_{3t}^{n,i} = 0, \quad if \ (||\mathbf{x}_t^{n,i}, \mathbf{x}_t^f|| < \Phi_d \quad \& \quad M_{\mathbf{X}_t^{n,i}} < \Phi_m), \quad i \in \{r, l\} \tag{4.14}$$

where $M_{\mathbf{X}_t^{n,i}}$ is the average movement under the sub-particle $\mathbf{x}_t^{n,i}$, Φ_m is the threshold for the amount of movement, and Φ_d is a threshold for the distance of the hand particle to the face particle and can be determined with respect to the face width. Equation 4.14 sets the likelihood to zero for the sub-particles where there is no head movement, which means that there is no hand in that region. Figure 4.3 shows the tracking results during a hand-face occlusion.

The last criterion is to use additional constraints on respective locations of the hands and the face. This criterion is needed to prevent wrong object assignments especially after occlusions. The assignment for the left and right hands can be exchanged when the hands separate after an occlusion. We prevent such situations by restricting the respective horizontal locations of the hand particles.

Let h_t^n be the difference of the horizontal coordinates of the hand sub-particles.

$$h_t^n = x_t^{n,l} - x_t^{n,r} \tag{4.15}$$

h_t^n is negative if the left hand of the signer is on the right of the right hand. We only allow hands to be crossed up to a certain horizontal distance by setting the likelihood

(a) (b) (c) (d)

Figure 4.4. Tracking with particle filters: (a) Original image, (b) Particle distribution with independent PFs, (c) Particle distribution with joint PF, (d) Particle distribution with semi-independent PFs

to zero if this negative distance exceeds a threshold:

$$\mathbf{z}_{4t}^{n} = 0, \; if \; (h_t^n < -1 \cdot \Phi_h) \tag{4.16}$$

Having defined all the partial likelihoods (Equations 4.12, 4.13, 4.14, 4.16), we calculate the joint likelihood by the multiplication of all the partial likelihoods (Equation 4.11). The joint likelihood is calculated for the objects that stay in the scene. If any of the objects disappears, it is excluded from the likelihood calculations. We assume an object has disappeared if all of the sub-particles of that object have zero weight:

$$\sum_{n} z_t^{n,i} = 0 \tag{4.17}$$

Note that the joint likelihood is no longer between zero and one and it should be normalized when setting the particle weights:

$$\pi_t^n = (\mathbf{z}_t^n - \min_n(\mathbf{z}_t^n))/(\max_n(\mathbf{z}_t^n) - \min_n(\mathbf{z}_t^n)) \tag{4.18}$$

4.2.5. The Joint PF

Figure 4.5 shows the pseudo-code of the joint PF. $\mathbf{x}_t^n$ is the joint object state, as defined in Equation 4.8. The first step is to determine the initial states and the weights of the particles, with respect to the prior distribution. The initial distribution of the particles are determined with respect to an explicit detection step based on connected component labeling. The particles at time t are determined by the re-sampling, prediction and weight setting steps. At the re-sampling step, new particles are sampled with replacement from the weighted particles at time $t-1$. At this step the weights of the new particles are equally assigned. The states of the re-sampled particles at time t are determined by the object dynamics, as defined in Equation 4.9 and by an additional mean-shift step. The weights are determined by normalizing the joint likelihood (Equation 4.11 and Equation 4.18).

1. Initialization: $\{(\mathbf{x}_0^n, \pi_0^n)\}_{n=1}^N$
2. For $t > 0$
   (a) Re-sampling:
   $\{(\mathbf{x}_{t-1}^n, \pi_{t-1}^n)\} \rightarrow \{(\mathbf{x'}_{t-1}^n, 1/N)\}$
   (b) Prediction: $\mathbf{x''}_t^n = f(\mathbf{x'}_{t-1}^n)$
   $\{(\mathbf{x'}_{t-1}^n, 1/N)\} \rightarrow \{(\mathbf{x''}_t^n, 1/N)\}$
   (c) Mean-shift iterations: $\mathbf{x}_t^n = MSh(\mathbf{x''}_t^n)$
   $\{(\mathbf{x''}_t^n, 1/N)\} \rightarrow \{(\mathbf{x}_t^n, 1/N)\}$
   (d) Weight setting: $\pi_t^n \propto \mathbf{z}_t^n = h(\mathbf{x}_t^n)$
   $\{(\mathbf{x}_t^n, 1/N)\} \rightarrow \{(\mathbf{x}_t^n, \pi_t^n)\}$, $\sum_{n=1}^N \pi_t^n = 1$
   (e) Estimation: $\hat{\mathbf{x}}_t = E[\{(\mathbf{x}_t^n, \pi_t^n)\}]$

Figure 4.5. Joint PF algorithm

4.3. Semi-Independent PF

For comparison purposes with the joint PF, we present a Semi-Independent (SI) PF, which follow a similar strategy as [91]. Once each PF determines the weights of its particles, we update those weights with respect to the estimates of other filters.

The weights of the left and right hands are updated with respect to their distance to the other hand's estimate (Equation 4.19) and also to the face estimate (Equation 4.20):

$$\pi_t^{n,i} = \pi_t^{n,i} \cdot (1 - 1/exp(\alpha \cdot ||\mathbf{x}_t^{n,i}, \hat{\mathbf{x}}_t^j||)) \tag{4.19}$$

$$\pi_t^{n,i} = 0,\ if\ (||\mathbf{x}_t^{n,i}, \hat{\mathbf{x}}_t^f|| < \Phi_d\ \ \&\ \ M_{\mathbf{X}_t^{n,i}} < \Phi_m) \tag{4.20}$$

where $i \in l, r$ and $i \neq j$.

To prevent wrong hand assignments, we switch the assignments for the hands if the horizontal distance between the estimates exceeds a threshold. It is calculated as in Equation 4.15, with the difference that the calculation is done using the estimates, not the particles themselves.

4.4. Experiments and Results

We evaluate the algorithm performance on manually labelled ground truth data of two videos from the broadcast news. The videos have a resolution of 210x248 pixels and a duration of 22008 frames, with 25 fps. We compare the tracked positions of the hands and the face with the ground truth positions and calculate the accuracy of correct tracking. We compare the normalized distance between the tracked and the ground truth positions, which is the ratio of the Euclidean distance between the two points divided by the shorter axis of the detected ellipse. If this normalized distance is greater than one, we consider this tracked position as false. The tracking accuracy for each object is calculated by dividing the number of correct frames for that object to the total number of frames.

Table 4.1 shows the comparison of two different PF approaches, joint and semi-independent, with or without using mean-shift with different number of samples. The results show that the joint PF approach with MSh has significantly higher accuracy than the semi-independent approach. Similar or better accuracies are obtained with MSh embedded method with substantially fewer number of particles. The best per-

Table 4.1. Comparison of SI and Joint PF approaches, w/ or w/o MSh, for different number of samples, in terms of tracking accuracy

		# Samples	Right H. (%)	Left H. (%)	Face (%)
joint	w/ MSh	50	94.49	96.24	99.98
joint	**w/ MSh**	**100**	**96.41**	**97.43**	**99.97**
joint	w/ MSh	200	96.37	97.09	99.97
joint	w/o MSh	100	66.73	72.34	58.17
joint	w/o MSh	200	79.17	84.97	77.77
joint	w/o MSh	500	84.66	89.84	84.17
SI	w/ MSh	50	77.58	89.55	99.97
SI	w/ MSh	100	82.91	91.85	99.97
SI	w/ MSh	200	83.74	90.57	99.97
SI	w/o MSh	100	72.90	81.57	100.00
SI	w/o MSh	200	72.96	82.41	100.00
SI	w/o MSh	500	79.99	87.27	100.00

forming method, considering the accuracies for the hands, is the joint PF with mean shift with 100 samples: the correct tracking accuracy is 99.9 for the face, 96.4% for the right hand and 97.4% for the left hand.

Detailed analysis for the best performing approach is shown in Table 4.2. The average distance (DIST) is calculated by the sum of Euclidean distances between the ground truth positions and the detected object center, divided by the total number of frames that the object exists. We also report number of false alarms (FA), missed objects (FR), wrong locations (WL), the total erroneous frames (TEF) and the tracking accuracy. TEF is the sum of FA, FR and WL.

We also analyze the performance of the tracking algorithm during occlusions. Figures 4.2 and 4.3 show the estimates of the algorithm during a hand-hand and hand-face occlusion, respectively. Table 4.3 shows the performance of the joint PF during occlusions between Right hand - Face (RF), Left hand - Face (LF) and Left - Right

Table 4.2. Comparison of joint PF results with ground truth data. The average distances (DIST), false alarms (FA), false rejects (FR), wrong locations (WL), the total erroneous frames (TEF) and the accuracy (% Correct) are reported

	DIST	FA	FR	WL	TEF	% Correct
Face	9.97	0	0	6	6	99.97
Right hand	6.44	198	29	564	791	96.41
Left hand	6.39	121	30	414	565	97.43

Table 4.3. Occlusion handling accuracy. The total number of occlusions and occlusion handling accuracy are reported for occlusions between Right hand-Face (RF), Left hand-Face (LF) and Left-Right hands (LR). We also report the average and maximum number of frames needed for recovery (FFR) after a failure

	Occlusions		FFR	
	Total	% Correct	Avg.	Max
RF	356	62.64	2.8	12
LF	1	100	-	-
LR	1943	93.62	1.6	18

hands (LR). We calculate the total number of frames with occlusion from the ground truth data, based on the distance between the objects. Results show that we handle the LR occlusions with 93.6% and RF with 62.6%. Note that hand-face occlusions can be particularly difficult to track and recovery time is important. The average number of frames needed for recovery (FFR) after a tracking failure during an occlusion is around 2-3 which shows the robustness of our algorithm to failures and its ability to recover fast.

4.5. Conclusions

We have presented a method for robust tracking of hands and the face on natural signing videos. The method is based on a joint particle filter approach and the joint likelihood models both the appearance and the respective positions of the hands and the face. This joint modeling enables us to accurately track occluding objects and to

maintain the correct labeling after the occlusion. The results show that our algorithm is also suitable for applications where tracking is needed for long durations, such as long sign sentences or sign conversations.

5. Recognizing Signs with both Manual and Non-Manual Components

5.1. Introduction

The problem of sign language recognition (SLR) is defined as the analysis of all components that form the language and the comprehension of a single sign or a whole sequence of sign language communication. SLR is a very complex task: a task that uses hand shape recognition, gesture recognition, face and body parts detection, and facial expression recognition as basic building blocks. Hand gesture analysis [16] is very important for SLR since the manual signals are the basic components that form the signs. However, without integrating non-manual signs, it is not possible to extract the whole meaning of the sign. In almost all of the sign languages, the meaning of a sign can be changed drastically by the facial expression or the body posture accompanying a hand gesture. Current multimodal SLR systems either integrate lip motion and hand gestures, or only classify either the facial expression or the head movement. There are only a couple of studies that integrate non-manual and manual cues for SLR [58].

Non-manual signs in sign language have only recently drawn attention for recognition purposes. Most of those studies attempt to recognize non-manual information independently, discarding the manual information. However, the analysis of non-manual signs is a must for building a complete SLR system: two signs with the same manual component and different non-manual components can have completely different meanings. The dependency and correlation of manual and non-manual information should also be taken into account in a recognition task. For the case of isolated signs, the manual and non-manual information coincide but the internal correlation and dependency are fairly low. For each isolated sign, there may or may not be a non-manual component. However, for continuous signing, the manual and non-manual components do not have to coincide synchronously and non-manual signs may cover more than one manual sign. This section is focused on isolated signs, assuming that subjects perform

a manual sign with or without an accompanying non-manual sign. Continuous signing is out of the scope of this work.

In this part of our work, we address the recognition of signs with both manual and non-manual components using a sequential belief-based fusion technique. We propose a methodology for integrating manual and non-manual information in a two-stage sequential approach. The manual components, which carry information of primary importance, are utilized in the first stage. The second stage, which makes use of non-manual components, is only employed if there is hesitation in the decision of the first stage. We employ belief formalism both to model the hesitation and to determine the sign clusters within which the discrimination takes place in the second stage.

The methodology is based on (1) identifying the level of uncertainty of a classification decision, (2) identifying sign clusters, i.e., groups of signs that include different non-manual components or variations of a base manual component, and (3) identifying the correct sign based on manual and non-manual information. Background information on hidden Markov models and belief functions, are given in the Appendix A and B respectively. Section 5.2 presents a numerical example on belief function usage. Section 5.3 explains the sequential belief-based fusion technique and our methodology to assign belief values to our decisions and to calculate the uncertainty. In Section 5.4, we compare our fusion technique with other state of the art fusion techniques and give the results of experiments with detailed discussions.

5.2. Belief Functions

Belief function formalism may be explained in many ways. The most intuitive way is to consider it as a generalization of probability theory which provides a way to represent hesitation and ignorance indifferently. This formalism is especially useful when the collected data is noisy or semi-reliable. The details and the background information on belief functions and the Partial Pignistic Transform (PPT) is given in Appendix B and more information can be obtained from [95, 96, 97, 98]. In this section we will illustrate the usage of belief functions on a numerical example (Table 5.1).

Table 5.1. Numerical example for belief function usage

	$\emptyset$	S	C	T	$\{S,C\}$	$\{S,T\}$	$\{T,C\}$	$\{S,C,T\}$
m_1	0	0.5	0	0	0.5	0	0	0
m_2	0	0	0	0	0	0	0.4	0.6
$m_{\textcircled{\cap}} = m_1 \textcircled{\cap} m_2$	0.2	0.3	0.2	0	0.3	0	0	0
M_1	0.2	**0.45**	0.35	0	0	0	0	0
$M_1\|\Omega$	0	**0.56**	0.44	0	0	0	0	0
M_2	0.2	0.3	0.2	0	0.3	0	0	0

Assume that we want to automatically classify a hand gesture. The gesture can be the trace of one of the following shapes: square (S), circle (C) or triangle (T). The gesture is analyzed by two different sensors, each giving an estimation of its shape. The observations of the sensors are expressed as beliefs, $m_1(.)$ and $m_2(.)$, and the powerset of hypotheses for the shape of the gesture is defined as $2^\Omega = \emptyset$, S, C, T, $\{S,C\}$, $\{S,T\}$, $\{T,C\}$, $\{S,C,T\}$. The two beliefs (m_1, m_2) are fused into a new belief $(m_{\textcircled{\cap}})$ via Dempster's rule of combination (Equation B.1), which is a rule to combine several belief functions into a global belief function. As the gesture has a single shape, the belief in union of shapes, $m_{\textcircled{\cap}}$, is meaningless from a decision making point of view. Then, if one needs to make a complete decision without hesitation (M_1), partial pignistic transform with the first frame of decision, 1-PPT, can be used (see Equations B.3 and B.4). In addition, if we assume that the analyzed gesture is always either of the three shapes we consider, then the belief in the empty set is not meaningful and conditioning on Ω leads to the same result as the classical pignistic transform $(M_1|\Omega)$. If the sensors are not precise enough to differentiate between two particular gestures, then, 2-PPT may be used (M_2). In this example, $M_2 = m_{\textcircled{\cap}}$ as there is no uncertainty in the elements with cardinality three to share: $m_{\textcircled{\cap}}(\{S,C,T\}) = 0$.

5.3. Sequential Belief Based Fusion

In a SLR problem, where each sign is modeled by a generative model, such as an HMM, the classification can be done via the maximum likelihood approach, where the

sign class of the HMM that gives the maximum likelihood is selected. However, this approach does not consider situations where the likelihoods of two or more HMMs are very close to each other. The decisions made in these kinds of cases are error-prone and further analysis must be made.

In HMM based SLR, each HMM typically models a different hypothesis for the sign to be recognized. Our purpose is to associate a belief function with these likelihoods. Then, it is possible to model these error-prone cases by associating high belief into the union of hypotheses. By analyzing the proportion of belief which is associated with multiple hypotheses, it is possible to decide whether the classification decision is certain or error-prone. Then, we propose the following process: If the analysis indicates significant uncertainty in the decision of the first classification step, a second classification step must be applied. This second classification step is only applied to classes among which the uncertainty is detected.

In the following sections, we explain how to convert the HMM likelihoods to beliefs, how to introduce uncertainty using the calculated belief values and how to use this uncertainty for sequential fusion.

5.3.1. Belief Function Definition from HMM Log-likelihoods

We present a method to derive a belief function over the powerset of classes from the HMM log-likelihoods calculated for each class. The purpose is to convert the information contained in the log-likelihoods into the evidential formalism, so that corresponding methods in data fusion and decision making are usable.

The general idea is to automatically copy the fashion in which an expert would associate a BF to a set of likelihoods: In the case of an HMM having a significantly larger likelihood than the other HMMs, an expert is able to place her/his belief into it. On the contrary, s/he is able to share the belief among a set of HMMs which roughly have the same likelihoods.

Calculate the HMM log-likelihoods for each class:

1. For all pairs, compute the difference in the log-likelihoods.
2. Compute the standard deviation of the differences.
3. For each pair, model the hesitation via a normal distribution.
4. For each pair, create a belief function based on the hesitation model. This BF defines the belief on each class separately and also on the combination of the classes.
5. Combine these belief functions into a single belief function.

Figure 5.1. Algorithm to compute belief function from HMM log-likelihoods

A way to understand this is to see the human being as capable of considering simultaneously several pairwise comparisons and to apprehend their global interactions. We propose to analytically copy this behavior: Let us consider all possible pairs of HMMs involved, and associate a belief function to each of them. We assume that the higher the likelihood is, the more believable the corresponding sign. Then, the margin between the likelihoods of two HMMs is a basis for local decision [6]. The next crucial step is to decide how to numerically associate a belief function over the powerset of every pair. In Figure 5.1, a summary of the algorithm to compute the beliefs from HMM log-likelihoods is given. The details of the algorithm is explained in B.2.

5.3.2. Introducing Uncertainty via Belief Functions

In our classification problem, we have signs with manual and non-manual information. We call signs that share the similar manual component as a "cluster". We hypothesize that it will be easier to differentiate between signs from different clusters. The signs in the same cluster can be differentiated via the non-manual component. However, the non-manual features can be noisy as a result of the data collection setup (i.e. 2D acquisition instead of 3D) and the feature extraction process. In some signs, the hand can be in front of the head, whereas in others, the face detector fails. Thus, we consider a potential absence of information with respect to this non-manual component. Then, our purpose is to make a decision only when it is sensible to do so, but also to accept a part of indecision when the information is not meaningful. We propose to apply the belief formalism to the problem,

- To model the hesitation among the gestures by a belief function computed from

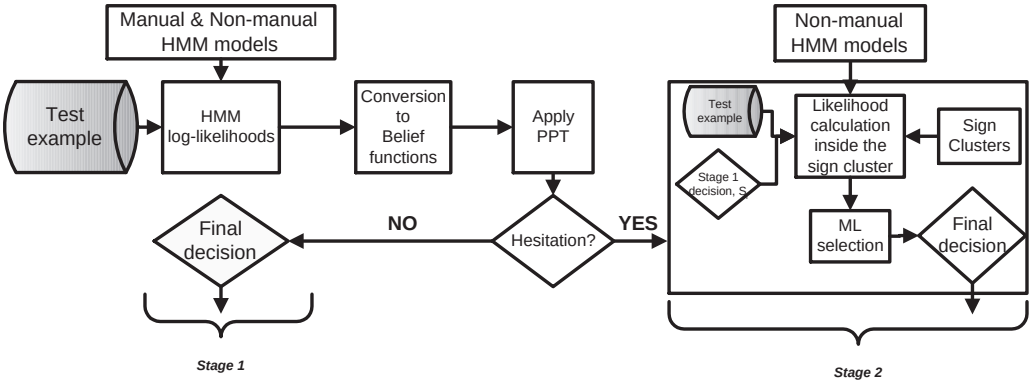


Figure 5.2. Sequential belief-based fusion flowchart

a set of likelihoods.

- To make a decision with the partial pignistic transform (Equation B.4).

With respect to the quality of the information available among the features, we assume that the decision between the clusters will be complete, but a hesitation may remain within a cluster (concerning the non-manual component). In order to make a final decision within a cluster, we need a second stage of decision.

5.3.3. Sequential Fusion with Uncertainty

The sequential belief based fusion technique consists of two classification phases where the second is only applied when there is hesitation (see Figure 5.2). The necessity of applying the second phase is given by the belief functions defined on the likelihoods of the first bank of HMMs. The eventual uncertainty calculated from those beliefs (via the PPT) is evaluated and resolved via the second bank of HMMs. In this setup, the assumption is that the HMMs of the first bank are more general models which are capable of discriminating all the classes up to some degree. The HMMs of the second bank are specialized models and can only be used to discriminate between a subset of classes, among which there is an uncertainty. These uncertainties between classes are used to identify the sign clusters in which the second bank of HMMs are capable of discriminating.

Table 5.2. Signs in eNTERFACE'06 Database

Base Sign	Variant	Hand Motion Variation	Head Motion (NMS)
Clean	Clean		
	Very clean		✓
Afraid	Afraid		
	Very afraid	✓	✓
Fast	Fast		
	Very fast		✓
Drink	To drink		✓
	Drink (noun)	✓	
Open (door)	To open		
	door (noun)	✓	
Here	[smbdy] is here		✓
	Is [smbdy] here?		✓
	[smbdy] is not here		✓
Study	Study		
	Study continuously	✓	✓
	Study regularly	✓	✓
Look at	Look at		
	Look at continuously	✓	✓
	Look at regularly	✓	✓

5.4. Methodology & Experiments

In order to assess the appropriateness of our belief-based method, we have performed experiments to compare it with several other mechanisms for fusing manual and non-manual signs. The experiments are conducted on a sign language database which has been collected during the eNTERFACE'06 workshop. In the following section, we give details about this database.

5.4.1. eNTERFACE'06 ASL Database

The signs in the eNTERFACE'06 ASL Database [2] are selected such that they include both manual and non-manual signs. There are eight base signs that represent words and a total of 19 variants which include the systematic variations of the base signs in the form of non-manual signs, or inflections in the signing of the same manual sign. A base sign and its variants will be called a "base sign cluster" for the rest of this paper. Table 5.2 lists the signs in the database. As observed from Table 5.2, some signs are differentiated only by the head motion; some only by hand motion variation and some by both. Two example signs are illustrated in Fig.5.3.

The dataset is divided to training and test sets where 532 examples are used for training (28 examples per sign) and 228 examples for reporting the test results (12

(a) (b)

Figure 5.3. Example signs from the eNTERFACE'06 ASL database: (a) HERE and NOT HERE, (b) CLEAN and VERY CLEAN

examples per sign). The subjects in training and test sets are different except for one subject whose examples are divided between training and test sets. The distributions of sign classes are equal both in training and test sets. For the cases where a validation set is needed, we apply a stratified 7-fold cross validation (CV) on the training set.

The details of the analysis and feature extraction on the eNTERFACE'06 ASL database are given in Section 7.1. Sign features are extracted both for manual signs (hand motion, hand shape, hand position with respect to face) and non-manual signs (head motion). For hand motion analysis, the center of mass (CoM) of each hand is tracked and filtered by a Kalman Filter. The posterior states of each Kalman filter: x, y coordinates of CoM, and horizontal, vertical velocity, form the hand motion features. Hand shape features are appearance-based shape features calculated on the binary hand images. These features include the parameters of an ellipse fitted to the binary hand and statistics from a rectangular mask placed on top of the binary hand. For head motion analysis, the system detects rigid head motions such as head rotations and head nods [99]. The orientation and velocity information of the head and the quantity of motion are used as head motion features.

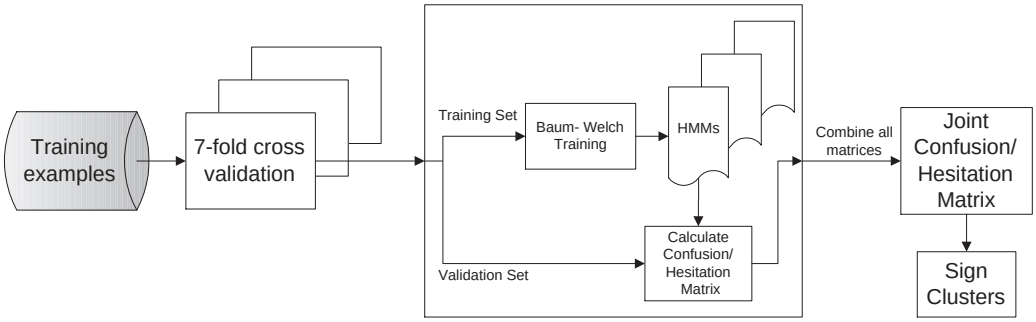


Figure 5.4. Identifying sign clusters by cross validation via confusion or hesitation matrices

5.4.2. Clustering for Sequential Fusion

In this context, we define a sign cluster as a group of signs which are similar and the differences are either based on the non-manual component or variations of the manual component. From the semantic interpretation of the signs in the database (see Table 5.2), we can define the base sign clusters as shown with the bold lines in Figure 5.5. In Figures 5.5(a) and (b), the bold lines indicate the semantic clusters.

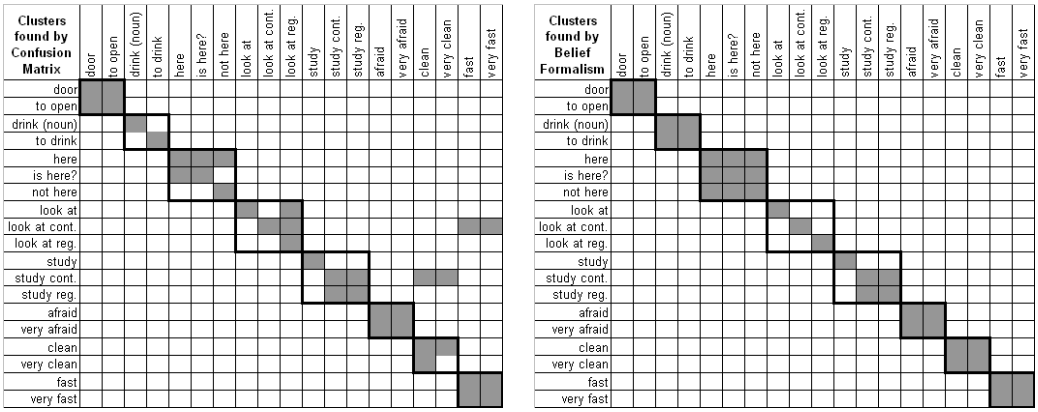
In a classification task, although one can utilize prior knowledge such as the sign clusters based on semantic information, this has some disadvantages. First, it is not guaranteed that these semantic clusters are suitable for the classification task, and second, the trained model will be database dependent and extending the database with new signs will require the re-definition of the cluster information. Thus, an automatic clustering method that depends on the data and considers the capabilities of the classifier would be preferable.

We propose two methods for automatic identification of the clusters: The first method is based on belief formalism, and the second method is based on the classification errors without any belief calculation. For the latter, we propose to use the confusion matrix for cluster identification; and for the former we propose to use the hesitation matrix. In both cases, the cluster identification is done by applying cross-validation on the training data. The confusion/hesitation matrices of each fold are combined to create a joint matrix, which is used to identify the clusters (Figure 5.4).

The joint confusion matrix is formed by summing the confusion matrices of the validation sets in a cross validation stage. We investigate the misclassifications by using the joint confusion matrix. To convert a joint confusion matrix to a sign cluster matrix, for each sign, we cluster all signs among which there is confusion. If all samples of a sign class are correctly classified, the cluster of that sign class only contains itself. Otherwise for each misclassification, we mark that sign as potentially belonging to the cluster. For example, assume that sign i is only confused with sign j. Then the sign cluster of class i is (i, j). The marked signs become members of the cluster when the number of misclassifications exceeds a given threshold. We use 10% of the total number of validation examples as the threshold value. Figure 5.5(a) shows the sign clusters identified via the confusion matrix for the eNTERFACE'06 sign data.

In belief formalism, we use the uncertainties in the decisions of the first stage classifier to define the sign clusters (Figure 5.5(b)). For this purpose, only the elements which are hesitation-prone are considered in a hesitation matrix. The elements without any hesitation, either complete mistakes or correct ones are excluded form the calculation of this hesitation matrix. This is equivalent to a confusion matrix but the sum over the matrix is equal to the number of hesitations. Each hesitation is multiplied by the number of elements among which the hesitation occurs. Then this matrix is transformed so that it is closed, transitive and reflexive. The classes of equivalence in this matrix correspond to the clusters. The number of elements within each hesitation is directly related to the γ parameter, which is the uncertainty threshold for the PPT. To simply tune it, we propose to approximately set it to the number of signs within the base sign clusters (i.e. $\gamma = 2$ or 3).

Figure 5.5 shows the sign clusters identified by the two techniques, by confusions in the joint confusion matrix and by uncertainties provided by the belief functions, respectively. Boldly outlined squares show the base sign clusters and for each row, shaded blocks show the identified clusters for the corresponding sign. The problem with the confusion matrix method is its sensitivity. Even a single mistake causes a cluster formation. On the other hand, the belief based method robustly identifies the uncertainties and provides robust clusters.



(a) (b)

Figure 5.5. Sign clusters (a) identified by the joint confusion matrix of 7-fold CV, (b) identified by the uncertainties between the classes in 7-fold CV. Clusters are shown row-wise, where for each sign row, the shaded blocks show the signs in its cluster. Bold lines show the clusters indicated by prior semantic knowledge, which we do not use

(a) (b)

Figure 5.6. Example signs: (a) DRINK, and (b) TO DRINK differ with the head motion and variations in the hand motion

At this point it is helpful to discuss the clustering results and their interpretation with respect to the signs in the database. As listed in Table 5.2, there are eight base signs in the database. The 19 signs are formed either by adding non-manual information to the base sign or by variations in the signing of the base sign and sometimes, both. Thus in a sign cluster, not all the confusions can be resolved by utilizing only non-manual information. Here we give the details of the base signs and discuss the clustering results shown in Figure 5.5(b).

- Signs DOOR and TO OPEN are only differentiated by the positioning of the hands and their speed and there is no non-manual information to differentiate between them. Although the clustering method puts these two signs in the same cluster, one can not expect to have a correct decision at the second step by only utilizing the non-manual component. For these signs the confusion must be

resolved at the first step by the manual information.

- Signs DRINK and TO DRINK are differentiated by both non-manual information and variations in signing (See Figure 5.6). TO DRINK sign imitates a single drinking action with the head motion. DRINK sign is performed without head motion and with repetitive hand motion. However, as the hand is in front of the mouth region, for some frames, the face detector fails and provides wrong feature values that mislead the recognizer.
- Signs HERE, IS HERE, NOT HERE have exactly the same manual sign but the non-manual sign differs. Thus when only manual information is used, confusions between these three signs are expectable. Non-manual information resolves the confusions in the cluster.
- For the LOOK AT sign, the differentiation is provided by both non-manual information and variations in signing. However, the hands can be in front of the head for many of the frames. For those frames, the face detector fails and provides wrong feature values that mislead the recognizer.
- The STUDY sign: It is interesting to observe that in Figure 5.5(b), the base study sign is clustered into two sub-clusters. This separation agrees with the nature of these signs: In the sign STUDY, there is a local finger motion without any global hand motion, and this directly differentiates this sign from the other two variations (See Figure 5.7). The confusion between the manual components of STUDY REGULARLY and STUDY CONTINUOUSLY can stem from a deficiency of the 2D capture system. The hand motion of these two signs differ mainly in depth. However, the non-manual components can be used at the second stage to resolve this confusion.
- For signs AFRAID, FAST, and CLEAN, a non-manual sign is used to emphasize their meaning (signs VERY AFRAID, VERY FAST, and VERY CLEAN). Each of these signs and their emphasized versions are put in the same cluster and the confusion inside these clusters can be resolved by utilizing the non-manual component.

(a) (b)

Figure 5.7. Example signs (a) STUDY, and (b) STUDY REGULARLY differ with the head motion and global hand motion

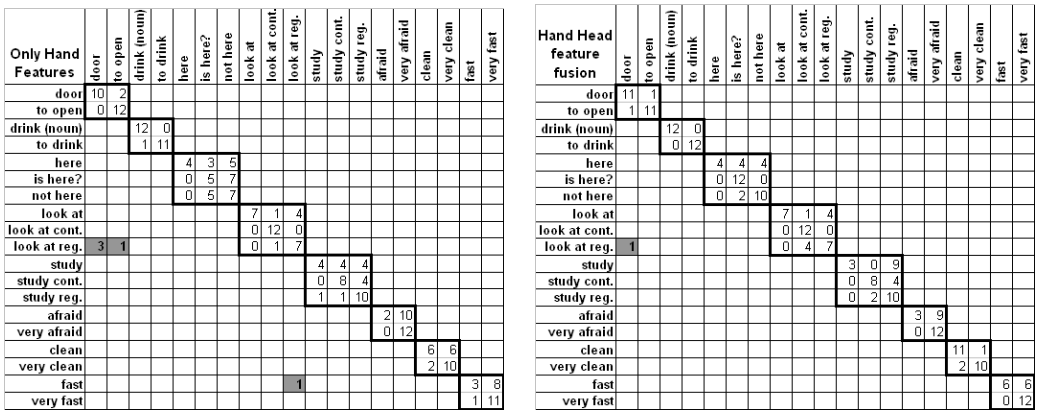
5.4.3. Reference Algorithms

To model the manual and non-manual components of the signs and perform classification, we train three different HMMs:

- HMM_M: Uses only manual features (hand motion, shape and position with respect to face)
- HMM_N: Uses only non-manual features (head motion)
- $\text{HMM}_{M\&N}$: Uses both manual and non-manual features (manual features plus head motion)

5.4.3.1. HMM Classification and Feature Level Fusion. We train HMMs for each sign and classify a test example by selecting the sign class whose HMM has the maximum log-likelihood. The HMM models are selected as left-to-right 4-state HMMs with continuous observations where Gaussian distributions with full covariance are used to model the observations at each state. The Baum-Welch algorithm is used for HMM training. Initial parameters of transition, prior probabilities and initial parameters of Gaussians are selected randomly.

We compare the classification performance of HMM_M and $\text{HMM}_{M\&N}$ to see the information added by the non-manual features via feature level fusion. The classification results of these two models should show us the degree of effective utilization of the non-manual features when combined into a single feature vector with manual features. Although there is no direct synchronization between the manual and non-manual components, the second model, $\text{HMM}_{M\&N}$, models the dependency of the two components



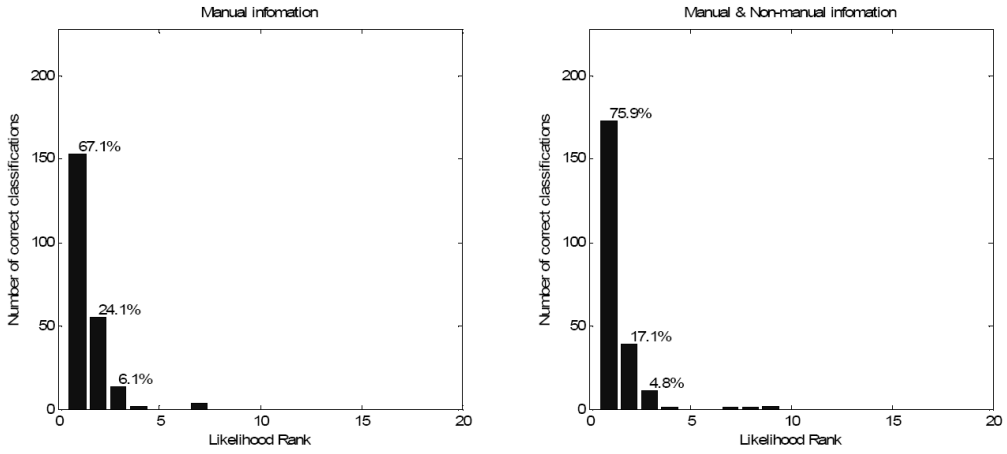
(a) (b)

Figure 5.8. Classification results and confusion matrices. (a) Confusion matrix of HMM_M. 97.8% base sign accuracy, 67.1% total accuracy. (b) Confusion matrix of $\text{HMM}_{M\&N}$. 99.5% base sign accuracy, 75.9% total accuracy. Rows indicate the true class and columns indicate the estimated class. Base sign and its variations are grouped and shown in bold squares. In both of the cases, classification errors are mainly between variations of a base sign

for sign identification.

The classification results and confusion matrices are shown in Figure 5.8. Although the classification accuracy of $\text{HMM}_{M\&N}$ is slightly better than HMM_M, total accuracy is still low. The high dimensionality of the feature vector (61 features per frame) can be a cause of this low accuracy. The curse of dimensionality affects HMM training. Another factor is that the non-manual features can be noisy as a result of wrong face detection, especially when hands are in front of the face. Besides, non-manual information is a secondary component of the sign and when analyzed together, the manual information may override the non-manual part. In any case, although it causes an improvement, non-manual information is not effectively utilized by feature fusion in HMM classification. However, it is worth noting that the classification errors in both of the models are mainly between variants of a base sign and out of cluster errors are very few.

A further investigation of the classification results shows that in about 97.5% of the examples, the true class resides among the first three highest likelihoods, if not the



(a) (b)

Figure 5.9. Rank distribution of the true class likelihoods of (a) HMM_M, 97.3% rank 3 accuracy, (b)$\text{HMM}_{M\&N}$, 97.8% rank 3 accuracy

maximum (see Figure 5.9). By further analyzing the first three highest likelihoods, one might increase the performance.

5.4.3.2. Parallel Score Level Fusion. Results of the previous section show that manual and non-manual information are not effectively utilized by feature level fusion. In parallel score level fusion, two independent experts are used and the scores (confidences, likelihoods ...) of these experts are combined with several combination rules. In this setup, the idea is to use one expert that models manual information (HMM_M) or manual and non-manual information together ($\text{HMM}_{M\&N}$) and to combine the scores of this expert with another one that models non-manual information (HMM_N). We use the sum rule to combine the log-likelihoods of the HMM experts. To use as a reference fusion method, we applied parallel fusion of the scores of (1) HMM_M and HMM_N, (2) $\text{HMM}_{M\&N}$ and HMM_N. The comparative results are shown in Table 5.3.

5.4.4. Sequential Score Level Fusion

Our proposed belief-based sequential fusion mechanism aims to identify the cluster of the sign in the first step and to resolve the confusion inside the cluster in the second step. For comparison purposes, we propose another sequential fusion technique which follows the same strategy but only uses the HMM likelihoods without any belief

formalism. In each of these techniques, information related to the sign cluster must be provided. In this section, we summarize each sequential fusion method.

5.4.4.1. Sequential Fusion based on HMM Likelihoods. A two step sequential fusion technique in which the sign clusters are automatically identified during training by the joint confusion matrix of 7-fold CV (see Figure 5.5(a)). In this method, the fusion methodology is based on the likelihoods of the HMMs. We give the base decision by a general model, $\text{HMM}_{M\&N}$, which uses both hand and head features in the same feature vector. The cluster of the selected sign is determined by the previously identified clusters. In the second stage the classifier only considers signs within the identified cluster. The decision is made by selecting the sign with the maximum likelihood of HMM_N.

The system uses the trained HMM models. The training is two-fold:

1. Train two HMM models for each sign, $\text{HMM}_{M\&N}$ and HMM_N.
2. Extract the cluster information via the joint confusion matrix of $\text{HMM}_{M\&N}$ (Section 5.4.2).

The fusion strategy for a test sample is as follows:

1. Calculate the likelihoods of $\text{HMM}_{M\&N}$ for each sign class and select the sign class with the maximum likelihood as the base decision.
2. Send the selected sign and its cluster information to HMM_N.
3. Combine the likelihoods of $\text{HMM}_{M\&N}$ and HMM_N with the sum rule and select the sign with the maximum likelihood as the final decision.

5.4.4.2. Sequential Fusion based on Belief Functions and Uncertainties. A two step sequential fusion technique in which the sign clusters are automatically identified during training by the uncertainties calculated from the beliefs on each sign is utilized (see Figure 5.5). The difference of this method from the likelihood-based one is twofold: (1)

the cluster identification method is based on belief functions (2) It is not mandatory to proceed to the second stage. If our belief about the decision of the first stage is certain, then we use that decision. The steps of the technique can be summarized as follows (also see Figure 5.2):

1. Automatically identify the clusters via uncertainties.
2. Define the uncertainty threshold (γ) for PPT and the standard deviation (σ) of the hesitation pattern.
3. Convert the likelihoods of the first stage HMMs ($\text{HMM}_{M\&N}$) to belief functions as explained in Appendix B.2.
4. Apply the PPT.
   (a) If there is no uncertainty in the result, decide accordingly.
   (b) Otherwise, identify the cluster of this sign and proceed to the second stage.
5. In the second stage, only consider the signs within the identified cluster.
6. The decision is made by selecting the sign with the maximum likelihood of HMM_N.

5.4.5. Results

The accuracies with different fusion techniques are summarized in Table 5.3. The automatically identified clusters using 7-fold CV can be seen in Figs. 5.5a and 5.5b, for the two techniques, fusion using likelihoods and fusion using belief functions, respectively. We have used the notations $\Rightarrow$ and $\rightarrow$, respectively for these two fusion methods to indicate the difference in the process of proceeding to the second stage, where $\Rightarrow$ indicates unconditional proceeding whereas $\rightarrow$ indicates a condition based on the belief-based analysis. Base sign clusters are as defined in the previous sections. Manually defined sign clusters are tuned manually by the human expert to emphasize the fact that even if it is tuned manually by taking the properties of the classification and analysis methods into consideration, the proposed method with automatically defined clusters is superior.

Sequential-belief based fusion has the highest accuracy (81.6%) among all im-

Table 5.3. Classification performance

	Models	Fusion method	Cluster identification	Test Accuracy
Reference	HMM_M	No fusion	-	67.1%
	$HMM_{M\&N}$	Feature	-	75.9%
	$HMM_M + HMM_N$	Parallel	-	70.6%
	$HMM_{M\&N} + HMM_N$	Parallel	-	78.1%
Proposed	$HMM_{M\&N} \Rightarrow HMM_N$	Sequential likelihood	Base sign clusters	73.7%
	$HMM_{M\&N} \Rightarrow HMM_N$	Sequential likelihood	Manually defined	78.1%
	$HMM_{M\&N} \Rightarrow HMM_N$	Sequential likelihood	Automatic via confusion matrix	75.0%
	$HMM_{M\&N} \Rightarrow HMM_N$	Sequential likelihood	Automatic via uncertainties	73.3%
	$HMM_{M\&N} \rightarrow HMM_N$	Sequential belief-based	Automatic via uncertainties	**81.6%**

plemented techniques. The reason for this success is both based on the robustness of the belief-based cluster identification and the possibility of accepting the first stage classification decision by taking advantage of belief formalism. The effect of the latter can be seen from the last two lines of Table 5.3, where the same clustering result, based on uncertainties calculated from belief functions, gives a very low accuracy if we do not apply belief-based decision analysis. For the former, when the clusters are not properly and robustly defined, the classification performance may degrade. This effect can be seen in Table 5.3 where we report the accuracies of sequential likelihood fusion with different cluster identification techniques. When compared with the base model, $HMM_{M\&N}$, the classification accuracy is lower in three of the cluster identification methods and only higher with manually defined clusters. The main reason is that when belief-based decision analysis is not applied, the classifier proceeds to the next stage unconditionally, regardless of the first stage decision, and correct classifications of the first step are altered.

Although the time dependency and synchronization of manual and non-manual features are not that high, feature fusion still improves the classification performance (13% improvement) by providing extra features of non-manual information. This improvement is also superior to parallel score fusion of manual and non-manual models,

showing the need for co-modeling. However, the modeling of $\text{HMM}_{M\&N}$ is not sufficient and still has low accuracy. The classification performance is improved by adding an extra expert, HMM_N, and by performing parallel score level fusion with another expert, $\text{HMM}_{M\&N}$ (3% improvement to $\text{HMM}_{M\&N}$ and 16% improvement to HMM_M). However, the sequential belief-based fusion and the clustering idea is superior since the manual information forms the primary component and the non-manual information forms the secondary component of a sign. The sequential belief-based fusion method processes the signs according to this information. The analysis of wrongly classified examples shows that in most of the cases the wrong decision is due to the lack of 3D information, or due to failure in face detection. Since we use 2D features, some of the signs resemble each other (e.g. signs STUDY CONTINUOUSLY and STUDY REGULARLY). Failures in face detection mainly occur when the hand occludes the face; thus resulting in wrong head tracking results. There are also a few examples where the wrong classifications are due to hand tracking mistakes.

5.5. Conclusions

We have compared several fusion techniques for integrating manual and non-manual signs in a sign language recognition system. A dedicated fusion methodology is needed to cover the specialties of the usage of manual and non-manual signs in sign languages. We propose to use a two-step fusion methodology: The first step mainly depends on the manual information and the second step only utilizes non-manual information. This is inspired by the fact that the manual component is the main component in sign language communication. In many of the signs, the information can be conveyed without the need for non-manual signals. However, when a non-manual sign is used, it may radically change the meaning of the manual sign, by either emphasizing it, or indicating a variation. Hence, one can not discard the non-manual signs in a complete recognition system. Our proposed belief-based fusion mechanism is based on this observation and applies a two-step fusion approach. The first step of the fusion mechanism applies feature level fusion on manual and non-manual components and attempts to make a decision. The decision is analyzed by the belief formalism and by considering a potential absence of information, and it can be accepted or a hesitation

between some of the sign classes can be expressed. This hesitation is expected to remain inside the sign cluster and the second step aims to resolve the hesitation by considering only the non-manual component.

The key point of our belief-based fusion approach is two-fold. First, it has a two-step decision phase and if the decision at the first step is without hesitation, the decision is immediately made, without proceeding to the next step. This would speed up the system, since there is no need for further analysis. Even in the case of a hesitation, the decision of the first step identifies the cluster which the test sign belongs to, if not the exact sign class. Second, the sign clusters are identified automatically at the training phase and this makes the system flexible for adding new signs to the database by just providing new training data, then training models for the new signs and running the belief formalism to find the new sign clusters.

These two key points root in the capability of the PPT to make a decision which is a balance between the risk of a complete decision and the cautiousness of a partial decision. It is able to provide a singleton decision when supported, but on the other hand, as long as the information is too hesitation-prone, it makes an incomplete decision. Then, it is automatically decided whether the second stage is used or not. Our results show that automatic belief based clustering even outperforms manual labeling based on semantic information in identifying base sign clusters and variations within clusters. Finally, this methodology can also be used in other linked problems, such as identifying grammatical processes or performance differences in sign languages provided that necessary features are extracted.

6. Combining Generative and Discriminative Models for Recognizing Hand Gestures and Signs

6.1. Introduction

Gesture and sign recognition relies on modeling the spatial and temporal components of the hand. This can be achieved by using models that can handle variable length sequences and the dynamic nature of the data. Several methods such as Finite State Machines (FSM), Time-delay neural networks (TDNN), HMMs or template matching [16] are applied in these kinds of systems, with HMMs being the most extensively used method.

Fisher kernels are proposed as a methodology to map variable length sequences to a new fixed dimension feature vector space [100]. The mapping is obtained by differentiating of the parameters of an underlying generative model. This new feature space is called the Fisher score space [101] and on this score space, any discriminative classifier can be used to perform a discriminative training. The main idea of Fisher kernels is to combine generative models with discriminative classifiers to obtain a robust classifier which has the strengths of each approach. Since each Fisher score space is based on a single generative model, the new feature space is assumed to be suitable for binary classification problems in nature.

In the literature, Fisher kernels have been applied to binary classification problems such as bio-sequence analysis [100], protein homology detection [102], and also to multi-class classification (multi-class classification) problems such as audio classification [103], speech recognition [101], object recognition [104], texture classification [105], and face recognition [106]. To solve these multi-class classification problems, in most of these works, the researchers apply either a one-versus-one (1vs1) or a one-versus-all (1vsAll) scheme. In 1vs1 and in 1vsAll, binary classification is applied to two classes and then the decisions of the binary classifiers are combined [103][101][104][105]. In [106], the authors concatenated all the Fisher scores generated from the models of each

class into a single feature vector. Then, they apply a multi-class classification to this combined feature vector and achieve higher recognition performance when compared to binary classification schemes. There are also several other approaches, other than Fisher kernels, to learn sequential data via discriminative models [107, 108].

This study aims to use Fisher kernels to map the original variable length gesture and sign sequences based on HMMs to the fixed dimension Fisher score space, and apply a discriminative multi-class classification on this new feature space. We propose a new multi-class classification scheme that applies a multi-class classification on the Fisher score space of each generative model. In this approach, we use the discriminative power of the Fisher scores of one class to classify other classes. We compare this scheme with the other techniques applied for multi-class classification and show that the method is both accurate in comparison with the binary classification schemes and computationally more effective than concatenating all the score spaces into one feature vector, especially in terms of the memory requirements for high dimensional problems. Our results show that if the multi-class scheme is not properly determined, the recognition performance of the combined classifier may decrease even with respect to the underlying generative model.

One disadvantage of the Fisher scores is the high dimensionality of the generated feature vectors. The dimension of the new feature vector is directly related to the number of parameters of the underlying generative model. Although the generative models for gesture and sign sequences are as simple as a left-to-right HMM with a few states, the dimensionality of the Fisher scores gets higher with the feature dimensionality of the sequences and the number of classes. So both the computation of the Fisher scores and the discriminative training on the new feature space become costly. We analyse the effect of the parameters of the generative model and each score space, on the recognition performance. To reduce this computational complexity, we propose and compare several methodologies such as parameter selection, dimensionality reduction with PCA or LDA, and score space selection. We show that the complexity can be further reduced, without any compensation in the accuracy, by an intelligent score space selection strategy.

We conduct experiments on two datasets, a hand gesture and a sign language dataset. We compared the performance of several multi-class classification schemes and show that the proposed scheme provides the highest accuracy and enhances the performance of the base classifier the most. We also performed experiments on reducing the computational complexity of the Fisher score computation and classification steps. The organization of the chapter is as follows: In Section 6.2, we introduce the Fisher kernel methodology in detail. The multi-class classification strategies are discussed in Section 6.3 and Section 6.4 presents our proposed multi-class classification strategy. In Section 6.5, we discuss several strategies for reducing the computational cost of Fisher score calculation and classification. The results of the experiments are reported in Section 6.6.

6.2. Fisher Kernels and Score Spaces

A kernel function can be represented as an inner product between feature vectors:

$$K(X_i, X_j) =< \phi(X_i), \phi(X_j) > \tag{6.1}$$

where ϕ is the mapping function that maps the original examples, X, to the feature vectors in the new feature space. By choosing different mapping functions, ϕ, one has the flexibility to design a variety of similarity measures and learning algorithms.

A mapping function that is capable of mapping variable length sequences to fixed length vectors enables the use of discriminative classifiers for variable length examples. Fisher kernel [100] defines such a mapping function and is designed to handle variable length sequences by deriving the kernel from a generative probability model. The gradient space of the generative model is used for this purpose. The gradient of the log likelihood with respect to a parameter of the model describes how that parameter contributes to the process of generating a particular example. All the structural assumptions encoded in the model about the generation process are naturally preserved in this gradient space [100]. Higher order Fisher Kernels can also be constructed by taking the second or third order derivatives.

Fisher Score, U_X, is defined as the gradient of the log likelihood with respect to the parameters of the model:

$$U_X = \nabla_\theta log P(X|\theta) \tag{6.2}$$

The unnormalized Fisher Kernel, U_X, defines a mapping to a feature vector, which is a point in the gradient space of the manifold of the probability model class. The direction of steepest ascent in $logP(X|\theta)$ along the manifold can be calculated by the Fisher Score, U_X. The local metric of this Riemannian manifold is given by the Fisher Information Matrix (also known as the Hessian Matrix), I, where $I = E(U_X U_X^T)$. By mapping the examples into feature vectors using $\phi(X) = I^{-\frac{1}{2}} U_X$, the normalized Fisher Kernel can be defined as follows [100].

$$K(X_i, X_j) = U_{X_i}^T I^{-1} U_{X_j}^T \tag{6.3}$$

The normalization via the Fisher Information Matrix is applied to obtain linear translation invariance. It is important to note that true normalization can only be obtained by using the full covariance matrix of the score space. However, the exact calculation is only possible when the underlying distribution is known. Another problem is the computational load of the inversion of large matrices. Approximations to this matrix must be made. Fisher Information Matrix is one of the approximations to the covariance matrix of the score space and this approximation is only valid when the expectation in the score space is 0. Another approximation is to use the diagonal of the covariance matrix estimated from the scores in training set. In some situations, where normalization is not essential, a simplified form of the Fisher Kernel can also be used.

$$K_U(X_i, X_j) = U_{X_i}^T U_{X_j}^T \tag{6.4}$$

In this work, we normalized the score space using the diagonal of the covariance matrix of the score space estimated from the training set.

Table 6.1. Fisher, likelihood and likelihood ratio score spaces

Score Space	Feature Vector
FSS	$\nabla_{\hat{\Theta}_1} log\ p_1(O\|\hat{\Theta}_1)$
LSS	$\begin{bmatrix} log\ p_1(O\|\hat{\Theta}_1) \\ \nabla_{\hat{\Theta}_1} log\ p_1(O\|\hat{\Theta}_1) \end{bmatrix}$
LRSS	$\begin{bmatrix} log\ p_1(O\|\hat{\Theta}_1) - log\ p_2(O\|\hat{\Theta}_2) \\ \nabla_{\hat{\Theta}_1} log\ p_1(O\|\hat{\Theta}_1) \\ -\nabla_{\hat{\Theta}_2} log\ p_2(O\|\hat{\Theta}_2) \end{bmatrix}$

In practice, Fisher Scores are used to extract fixed size feature vectors from variable length sequences modeled with any generative model. This new feature space can be used with any discriminative classifier. However, the dimensionality of this new feature space can be high when the underlying generative model has many parameters and the original feature space is multivariate. Thus, SVMs become a good choice of a classifier since they do not suffer from the curse of dimensionality.

6.2.1. Fisher Score Spaces

Score spaces are generalizations of Fisher kernels and define the mapping space [109]. A score space is derived from the likelihood of generative models. Score vectors are calculated by applying a score operator to the score argument. Score argument can be the log likelihood or posterior of the generative model. Score operator can be the first or second derivative, or the argument itself. Table 6.1 shows the Fisher (FSS), Likelihood (LSS) and Likelihood Ratio (LRSS) score spaces. $p_1(O|\hat{\theta}_1)$ and $p_2(O|\hat{\theta}_2)$ are the likelihood estimates produced by the generative models of class 1 and class 2, respectively. Other score spaces and their derivations can be found in [101].

The difference between the FSS and the LSS is that the latter also uses the likelihood itself in the score vector. LRSS represents the two classes by putting the

likelihood ratio instead of the likelihood in the score vector, together with the score operators for each of the classes.

6.2.2. Fisher Kernels for HMMs Using Continuous Density Mixture of Gaussians

In gesture recognition problems, HMMs are extensively used and have proven successful in modeling hand gestures. Among different HMM architectures, left-to-right models with no skips are shown to be superior to other HMM architectures [110] for gesture recognition problems.

In this work, we have used continuous observations in a left-to-right HMM with no skips. The parameters of such an architecture are, prior probabilities of states, π_i, transition probabilities, a_{ij} and observation probabilities, $b_i(O_t)$ which are modeled by mixture of M multivariate Gaussians:

$$b_i(O_t) = \sum_{m=1}^{M} w_{im}\, \mathcal{N}(O_t; \mu_{im}, \Sigma_{im}) \tag{6.5}$$

where O_t is the observation at time t and w_{im}, μ_{im}, and Σ_{im} are weight, mean and covariance of the Gaussian component m at state i, with a total of M Gaussian components.

For a left-to-right HMM, the prior probability matrix is constant since the system always starts with the first state with $\pi_1 = 1$. Moreover, using only self-transition parameters is enough since there are no state skips ($a_{ii} + a_{i(i+1)} = 1$). Observation parameters in the continuous case are weight, w_{im}, mean, μ_{im} and covariance, Σ_{im} of each Gaussian component. The first order derivatives of the log-likelihood, $P(O|\theta)$

with respect to each parameter are given below:

$$\nabla_{a_{ii}} = \sum_{t=1}^{T} \frac{\gamma_i(t)}{a_{ii}} - \frac{1}{T\ a_{ii}\ (1-a_{ii})} \tag{6.6}$$

$$\nabla_{w_{im}} = \sum_{t=1}^{T} [\frac{\gamma_{im}(t)}{w_{im}} - \frac{\gamma_{i1}(t)}{w_{i1}}] \tag{6.7}$$

$$\nabla_{\mu_{im}} = \sum_{t=1}^{T} \gamma_{im}(t)\ (O_t - \mu_{im})^T\ \Sigma_{ik}^{-1} \tag{6.8}$$

$$\nabla_{\Sigma_{im}} = \sum_{t=1}^{T} \gamma_{im}(t)\ [-\Sigma_{im}^{-1} + \Sigma_{im}^{-1}\ (O_t - \mu_{im})\ (O_t - \mu_{im})^T\ \Sigma_{im}^{-1}] \tag{6.9}$$

where $\gamma_i(t)$ is the posterior of state i at time t and $\gamma_{im}(t)$ is the posterior probability of component m of state i at time t, and T is the total length of the data sequence. Since the component weights of a state sum to 1, one of the weight parameters at each state, i.e. w_{i1}, can be eliminated. The derivations of these gradients are given in A.2. These gradients are concatenated to form the new feature vector which is the Fisher score. The log-likelihood score space where log-likelihood itself is also concatenated to the feature vector is given as:

$$\phi_{O_t} = diag(\Sigma_S)^{-\frac{1}{2}} \left[\ log\ p(O_t|\theta)\ \nabla_{a_{ii}}\ \nabla_{w_{im}}\ \nabla_{\mu_{im}}\ \nabla_{vec(\Sigma)_{im}}\ \right]^T \tag{6.10}$$

When the sequences are of variable length, it is important to normalize the scores by the length of the sequence. We have used *sequence length normalization* [109] for normalizing variable length gesture trajectories by using normalized component posterior probabilities, $\hat{\gamma}_{im}(t) = \frac{\gamma_{im}(t)}{\sum_{t=1}^{T} \gamma_i(t)}$, in the above gradients.

6.3. Methods for Multi-class Classification Using Fisher Scores

As Fisher kernels are extracted from generative models which are trained with the examples of a single class, the new feature space of Fisher scores is mainly representative of the examples of that class. In [100], where the idea of Fisher kernels is proposed, the authors applied Fisher scores to a binary classification problem.

For a binary classification problem, one might have three different score spaces based on likelihoods:

1. LSS from the generative model of Class 1
2. LSS from the generative model of Class 2
3. LRSS from the generative models of Class 1&2

LRSS contains discriminative features from each class and provides a good representation for binary classification problems. Thus, it gives slightly better results [101] with respect to LSS, which is based on single class information.

Similarly, for a multi-class problem, Fisher scores from each generative model must be combined to obtain a good multi-class representation. In the next section, we summarize four schemes that are commonly used for the multi-class classification on Fisher scores. We present our approach, (M_{DLC}) in Section 6.4. A summary of all the schemes is given in Figure 6.1.

6.3.1. Commonly Used Multi-Class Classification Methods

To extend the Fisher scores to multi-class classification problems, a general method is to apply binary classifications and combine the results via decision level combinations. The first three schemes, B_{1vs1}, B_{1vs1R}, and B_{1vsALL} are the commonly used multi-class classification techniques based on a binary classifier, such as 1vs1 and 1vsAll. Alternatively, in [106] authors use a feature level combination approach (M_{FLC}) and concatenate all the Fisher scores into a single feature vector. Then, they apply multi-class classification on the combined feature vector.

6.3.2. $\mathbf{B_{1vs1R}}$: One-vs-One Classification based on LRSS

Since the likelihood ratio score space is the best performing score space for binary problems, a multiclass scheme which performs a binary classification for each pair of classes, (i, j), by using the LRSS of classes i and j can be formed.

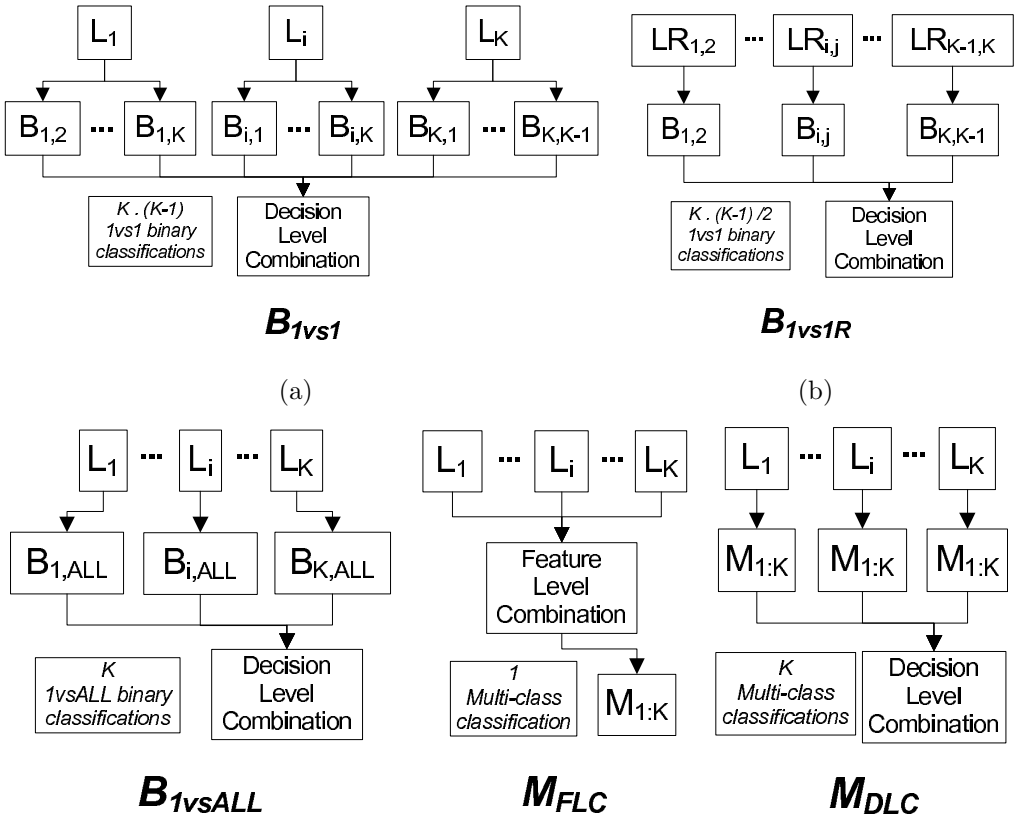


Figure 6.1. Multiclass classification strategies: (a) B_{1vs1}, (b) B_{1vs1R}, (c) B_{1vsALL}, $(d)M_{FLC}$, and (e) M_{DLC}

For each class pair (i, j), a binary classification is performed to classify whether the example belongs to class i or j. The binary classifier for class pair (i, j) uses the LRSS of classes (i, j). Note that the LRSS of (i, j) is the same as (j, i), except for a sign difference.

For a problem of K classes, $\frac{K \cdot (K-1)}{2}$ binary classifiers must be trained with examples of class i and j for each (i, j) pair, where $i, j = 1 \ldots K$ and $i \neq j$ since i-vs-j and j-vs-i are symmetric problems. A test example is given to each trained classifier. The final result is obtained by weighting each classifier by its posterior probability, combining the weighted votes, and selecting the class with the maximum vote.

6.3.3. B_{1vs1}: One-vs-One Classification based on LSS

Instead of using LRSS, we can utilize the LSS of each class: For each class pair (i, j), a binary classification is performed to classify whether the example belongs to class i or j. Note that the binary classifier for class pair (i, j) uses the LSS of class i and the binary classifier for class pair (j, i) uses the LSS of class j.

Unlike the previous scheme, (i, j) and (j, i) are not symmetric problems since each pair uses a different score space. Thus, for a problem of K classes, $K \cdot (K-1)$ binary classifiers must be trained with examples from class i and j for each (i, j) pair, where $i, j = 1 \ldots K \; and \; i \neq j$. A test example is given to each trained classifier. The final result is obtained by weighting each classifier by its posterior probability, combining the weighted votes, and selecting the class with the maximum vote.

6.3.4. B_{1vsALL}: One-vs-All Classification based on LSS

Another multiclass scheme can be formed by using one-vs-all methodology. For each class i, a binary classification is performed to classify whether the example belongs to class i or one of the other classes. The binary classifier for class i uses the LSS of class i.

For a problem of K classes, K binary classifiers must be trained with all the examples of the training set where for each classifier $C_i, i : 1 \ldots K$, examples of class i are labeled as 0 and all other examples are labeled as 1. A test example is given to each trained classifier. The final result is obtained by weighting each classifier by its posterior probability, combining the weighted votes, and selecting the class with the maximum vote.

6.3.5. M_{FLC}: Multiclass Classification based on Feature Level Combination of LSS

In this scheme, the score spaces of each class are combined into a single feature space and a multiclass classification is performed on this new feature space, as proposed in [106]. The main disadvantage of this scheme is the memory consumption since the resulting feature vector is the combination of multiple Fisher scores. When the input dimensionality, the number of parameters of the generative models and the number of classes are high, the dimensionality of the combined feature space will be extremely high, making it hard, or sometimes impossible, to keep the training data in the memory.

6.4. A New Multi-Class Classification Scheme for Fisher Scores

We propose a new strategy, M_{DLC}, which applies a multi-class classification on the LSS of each class and then combines the decisions of each classifier. M_{DLC} is especially suitable for applications where the number of classes is large and computational resources are critical.

M_{DLC} uses the Fisher scores of each class for the discrimination of all the other classes, not just for the class that produces the scores. In the above schemes, for a binary classification between class i and class j, Fisher scores extracted for related generative models (models for class i or j) are used. However, we show that Fisher scores extracted from class i may provide a discrimination for classes other than i (i.e. discrimination between class j and k).

For a problem of K classes, we train K multi-class classifiers with all the examples of the training set, using the original class labels. The main difference of this scheme is that, each of the K classifiers is performing a multi-class classification, whereas in the above schemes, except M_{FLC}, each classifier is a binary classifier.

The decisions are combined via weighted voting: a test example is given to each classifier and the final result is obtained by selecting the class with the maximum

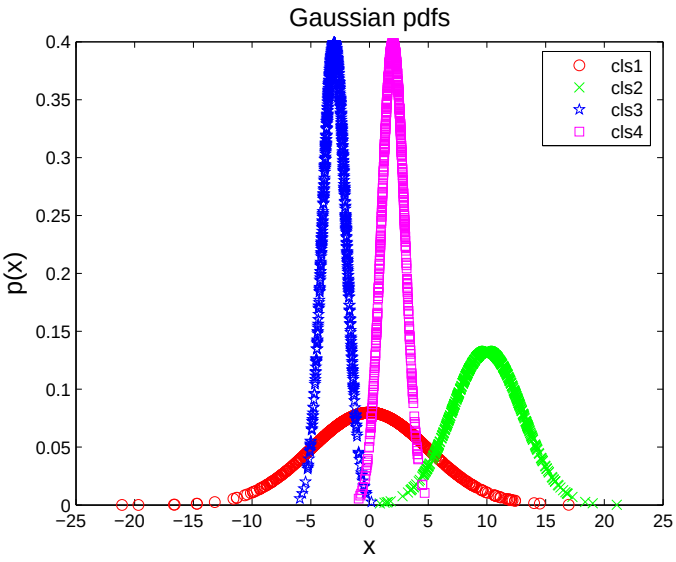


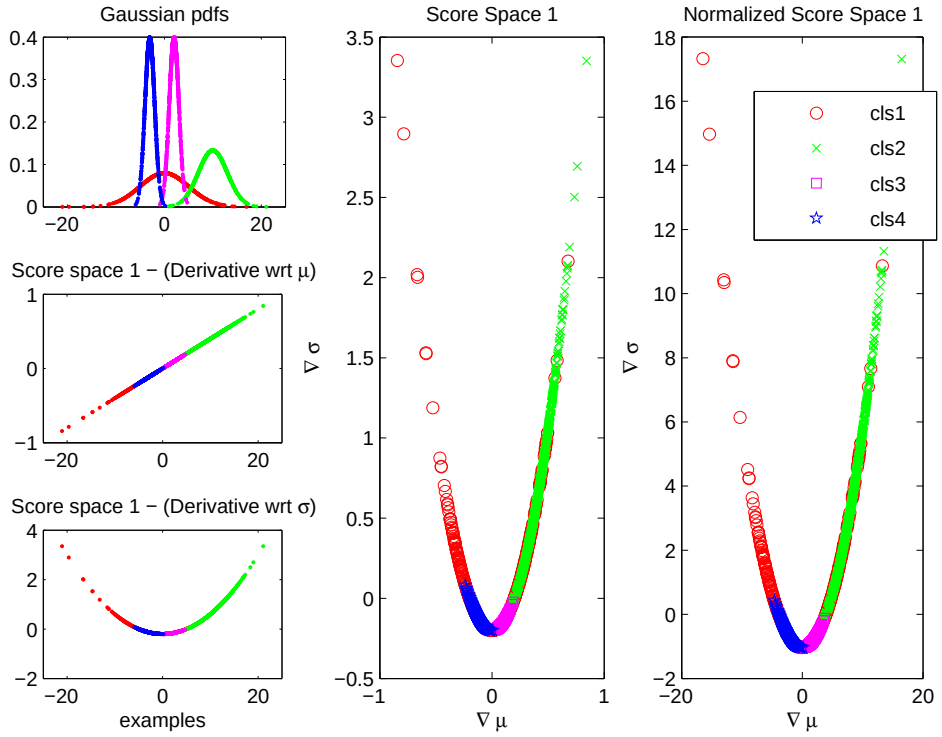
Figure 6.2. Artificial data generated from four Gaussian distributions

weighted vote, with posterior probabilities as the weights.

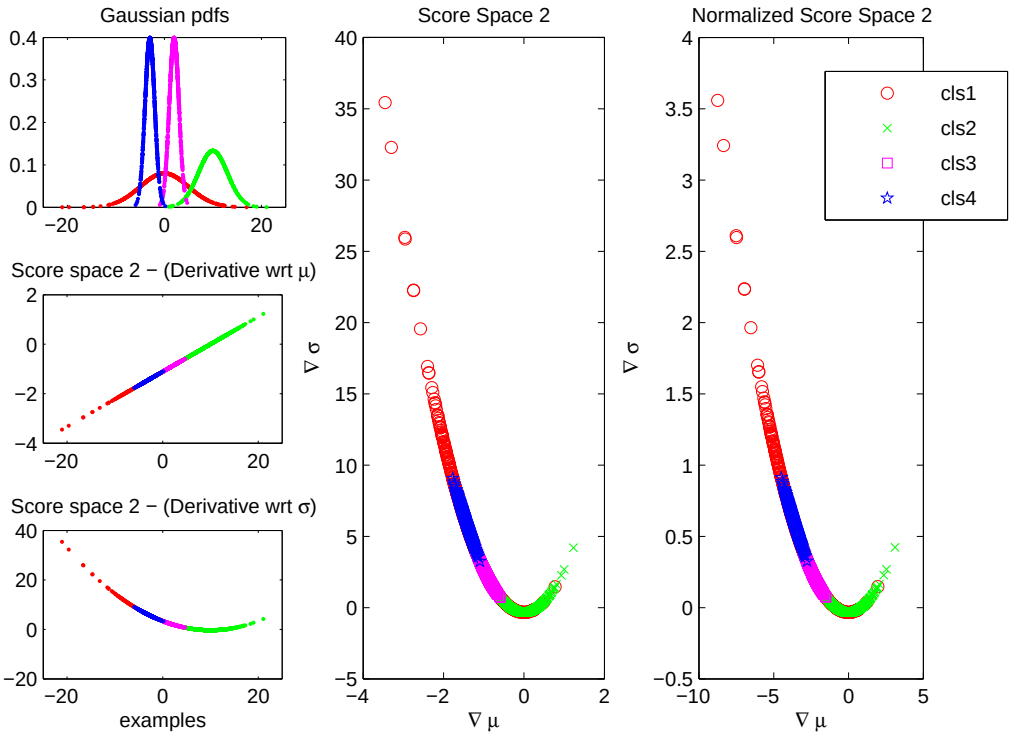
To demonstrate this multiclass scheme, let us consider a toy problem. We have generated random data from four different Gaussian distributions (Figure 6.2):

- Class 1: $\mathcal{N}(0, 5)$
- Class 2: $\mathcal{N}(10, 3)$
- Class 3: $\mathcal{N}(2, 1)$
- Class 4: $\mathcal{N}(-3, 1)$

Fisher scores are extracted for each Gaussian model for the parameters (μ, σ). The plots for score spaces can be seen in Figures 6.3 and 6.4 for all four classes. The classification results of applying a multi-class classification on each score space are shown in Table 6.2. Different rows show the score space used and the columns show the classification accuracy among all the classes and also of each class pair. The classification on each score space is performed with SVMs that apply multi-class classification.

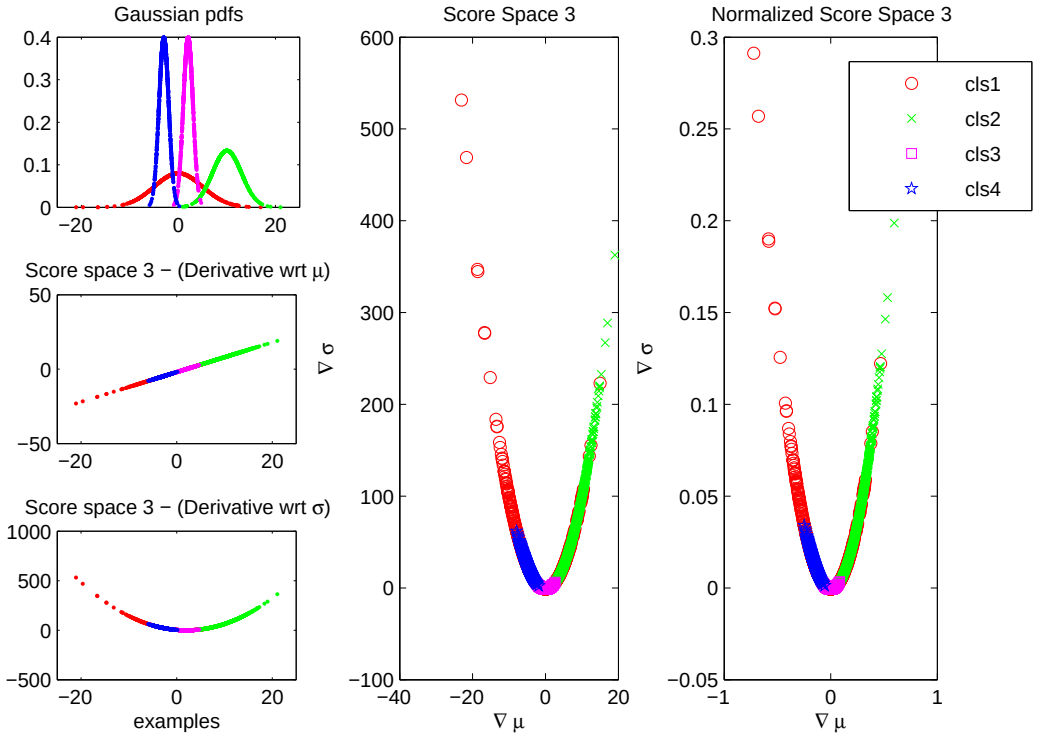


(a)

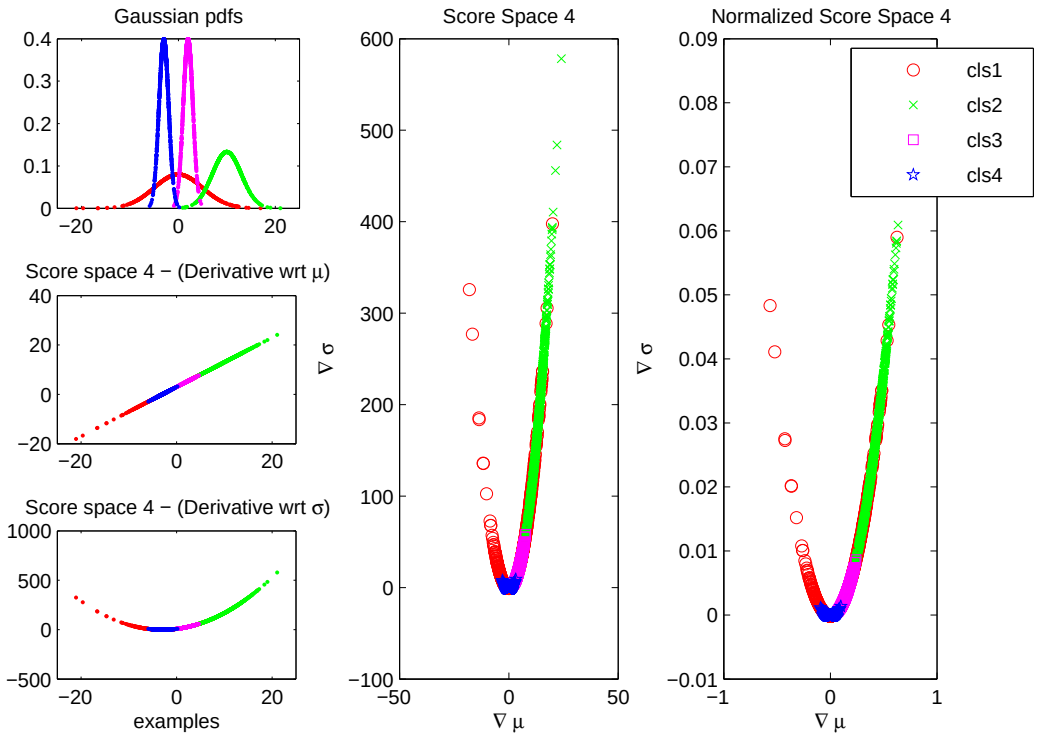


(b)

Figure 6.3. Score space plot of (a) Class 1, $N(0, 5)$ (b) Class 2, $N(10, 3)$



(a)



(b)

Figure 6.4. Score space plot of (a) Class 3, $N(2,1)$ (b) Class 4, $N(-3,1)$

Table 6.2. Classification results of applying multi-class classification on each score space

	Classification Accuracy (%) for						
Score Space(s) used	All classes	Classes 1&2	Classes 1&3	Classes 1&4	Classes 2&3	Classes 2&4	Classes 3&4
LLS_1	**79.25**	65.00	65.00	64.83	93.67	93.50	**93.50**
LLS_2	74.58	56.17	57.33	56.33	92.83	91.83	93.00
LLS_3	78.50	65.83	60.83	65.33	91.67	**96.17**	91.17
LLS_4	78.42	65.83	**68.83**	62.17	**94.67**	88.00	91.00
LLS_{1234}	79.17	**68.00**	68.17	**66.50**	91.83	90.17	90.33

Classes 2, 3 and 4 are easily separated by any of the score spaces, with LLS_1 providing the best discrimination among classes. As can also be seen from the distribution of the classes (Figure 6.2), using only the generative model of class 1, one can obtain very high classification performance. This small experiment shows that the score space obtained from the generative model of class i is not only capable of discriminating between class i and other classes, it also provides valuable discriminative information for classes other than i, such as j and k. For example, in Table 6.2, we observe that the score space of class 3, LLS_3, provides the highest accuracy in discriminating classes 2 and 4.

6.5. Reducing the Computational Cost

In this section we first list the elements that affect the computational cost and then discuss possible techniques that can be used to reduce it. The computational complexity of Fisher score usage arises from two phases: calculation of Fisher scores and classification.

In the calculation phase, critical parameters are the number of parameters of the underlying generative model, the length of the input data sequence, and the number of classes/models. In the classification phase, the Fisher score dimensionality (dependent

on the number of parameters of the generative model) and the number of classes/models are the critical parameters.

The computational cost of the calculation phase can be decreased by either using simpler base models and simpler feature vectors or by considering only a subset of the parameters of the base model for Fisher score calculation. For the rest of this section, we will concentrate on reducing the complexity of processes directly related to the Fisher score calculation and classification and assume that the models and the features of the gesture sequences are fixed.

One way to reduce the complexity is to select a subset of the model parameters. This effects both the Fisher score calculation time and the classification time since dimensionality of the Fisher score spaces will be smaller. Some of the parameters may have a greater effect on the classification performance and the optimal reduced parameter set with an acceptable performance can be determined in a training phase.

For the proposed M_{DLC} strategy, we can obtain classification decisions for all classes even with a single Fisher score mapping. Similar to the parameter selection, a score space selection strategy can be applied to find a reduced set of score spaces that performs well enough. This selection will affect both the Fisher score calculation and the classification time since the number of score spaces to work on will be smaller.

Finally, the classification time can be further reduced by applying dimensionality reduction techniques, such as PCA and LDA, on the calculated Fisher scores.

6.5.1. Principal Component Analysis

Principal Component Analysis (PCA) [111] seeks components that maximize the variance in the feature space. It is an unsupervised method, meaning that it does not use the class labels. The method performs an orthogonal linear transform to a new coordinate system where the variance is maximized. This new coordinate system is assumed to be useful for representing the data.

6.5.2. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) [111] is a supervised method that transforms the feature space to a new coordinate system which best discriminates among classes. It creates a linear combination of the features to maximize the distances between classes. In order to apply LDA and prevent singularity conditions in the formulas, one must have enough number of examples in the dataset. If the number of examples in the dataset is less than the input dimensionality, the dimensionality must be reduced by another technique, such as PCA, and LDA must be applied in this reduced feature space.

6.5.3. Score Space Selection Strategies

In a multi-class classification task with Fisher scores, multiple score spaces are obtained from the generative model each class. In M_{DLC} strategy, as each score space is able to provide a decision for all of the classes, a subset of these score spaces can be used instead of using all of them. Then one should select the best subset that gives the highest recognition accuracy. As the number of classes increases, the number of possible subsets increases exponentially. Hence, efficient and effective search techniques should be applied. We will use and compare the following search techniques, which are commonly used in the literature:

Selecting the best performing N score spaces. Order the score spaces with respect to their accuracy and select the first N as the subset.

Sequential Floating Forward Search. Sequential Forward Floating Selection (SFFS) [112] is originally proposed as a feature selection algorithm that applies a top down search strategy. At each iteration of SFFS, the algorithm attempts to add one of the features to the combined feature set, which is initially empty. The feature to be added is selected such that, when added to the combined set, it increases the accuracy the most.

This step is called the forward optimization or the inclusion. The floating property comes from the fact that after each forward optimization, a backward optimization (exclusion) step, which attempts to remove one of the features from the combined feature set, is applied. The feature to be removed is selected such that, when removed from the combined set, it increases the accuracy the most. The algorithm stops when there is no improvement in the accuracy at the end of forward and backward optimization steps.

We use SFFS as a score space selection method, where each "score space" is a classifier trained using the score space obtained from the generative model of each class. Our aim is to select the classifiers (score spaces) to be combined by running Sequential Forward Floating Selection (SFFS) with the classification accuracy as the objective function.

1. *Initialization:* Start with the empty set of combined score spaces, $\Omega_{comb} = \emptyset$. Set the unused score spaces subset to all score spaces: $\Omega_{unused} = \{ss_1, \cdots, ss_n\}$;
2. *Forward optimization:* For each score space, $ss_i \in \Omega_{unused}$, add this score space to Ω_{comb} to obtain the candidate subset, $\Omega_{cand} = \Omega_{comb} \bigcup ss_i$. Select the candidate subset and the score space which produces the highest accuracy, Ω^*_{cand} and ss^*_i, respectively. If the accuracy of Ω^*_{cand} is higher than the accuracy of Ω_{comb}, then set $\Omega_{comb} = \Omega^*_{cand}$ and $\Omega_{unused} = \Omega_{unused} - ss_i$
3. *Backward optimization:* For each score space, $ss_i \in \Omega_{comb}$, remove this score space from Ω_{comb} to obtain the candidate subset, $\Omega_{cand} = \Omega_{comb} - ss_i$. Select the candidate subset and the score space which produces the highest accuracy, Ω^*_{cand} and ss^*_i, respectively. If the accuracy of Ω^*_{cand} is higher than the accuracy of Ω_{comb}, then set $\Omega_{comb} = \Omega^*_{cand}$ and $\Omega_{unused} = \Omega_{unused} \bigcup ss_i$.
4. Try the forward and backward optimization steps successively until there is no performance improvement. Output the subset Ω_{comb}.

Sequential Floating Backward Search. Sequential Forward Backward Selection (SFBS) is a variant of SFFS, in which, instead of starting with an empty set, the

algorithm starts with all the features in the combined set and attempts to remove the features at each iteration. Thus SFBS first performs a backward optimization step, followed by a forward optimization step.

6.6. Experiments

We have performed experiments on two different databases. A two-handed gesture database collected at IDIAP [82] and an American Sign Language dataset collected during the eNTERFACE'06 workshop [2].

For the IDIAP database, following the hand detection and segmentation, we track the hand with a Kalman Filter and use Kalman Filter estimates of hand position, (x, y) in each frame (two features per hand per camera). Hand shape is modeled with a simple ellipse and the lengths of the ellipse axes and the rotation angle, (a, b, θ), are used as hand shape features (three features per hand per camera). The feature dimensionality is 20 per frame. More information on feature extraction can be found in [8].

For the eNTERFACE ASL database, we concentrated on the manual component for this work and extracted features only from the hand motion, shape and position. Note that some signs can only be differentiated by the non-manual components and the performance is expected to be low. The videos are first processed for hand and face detection and segmentation. Then, sign features are extracted for manual signs (hand motion, hand shape, hand position with respect to face). For hand motion analysis, the center of mass (CoM) of each hand is tracked and filtered by a Kalman Filter. The posterior states of each Kalman filter, x, y coordinates of CoM and horizontal, and vertical velocity are the hand motion features. Hand shape features are appearance based shape features calculated on the segmented hand images. These features include the width, height and orientation parameters of an ellipse and seven Hu moments [7] calculated on the binary hand image. The hand position is calculated with respect to the face center of mass. The horizontal and vertical distance of each hand CoM to the face CoM is used as hand position features. As a result, the feature vector dimensionality is 32 per frame (four hand motion, 10 hand shape, two hand position

features for each hand).

6.6.1. Comparison of Multiclass Strategies

For both of the databases, we trained a left-to-right continuous HMM for each sign. Therefore, 7 and 19 HMMs are trained for IDIAP and eNTERFACE databases respectively. Each HMM has four states, and a single Gaussian density is used in each state. Fisher Score spaces are calculated for each HMM and a SVM classifier with an RBF or a linear kernel is used for classification. For each strategy, the kernel type and the parameters are determined separately by cross-validation over a set of parameter values.

Comparison of the different multi-class strategies, together with the performance of the underlying generative model on IDIAP and eNTERFACE databases are given in Table 6.3. The baseline accuracies, obtained by HMMs, of these two databases are 99% and 68.2% respectively. IDIAP is a small and easy dataset. The eNTERFACE database is more difficult and performance figures are in the 60-70% range as expected when only manual features are used. The advantage of using all score spaces to discriminate each class pair can clearly be seen from the classification accuracies where M_{DLC} outperforms all other binary schemes. Instead of using binary classifiers, using multi-class classifiers on each score space increases the accuracy and provides a better strategy for multi-class classification. Although M_{FLC} performs well on the IDIAP database, on the eNTERFACE database the computation can not be completed as a result of the huge memory requirement. For the IDIAP database, the baseline accuracy is already very high and the significances between the different schemes may not be apparent as a result of the ceiling effect. Despite the ceiling effect, there is around 0.6% increase in the accuracy with lower or equal standard deviation. Results on the eNTERFACE database show similar behavior with more significant differences between strategies: 5% increase in the accuracy with multi-class classification of Fisher Score Spaces. In both of the databases, the accuracy drops down with binary classifiers and it is possible to beat the base classifier accuracy only with multi-class classifiers.

Table 6.3. Comparison of different multi-class schemes on IDIAP and eNTERFACE databases. Average test accuracies of 10-fold cross validation ((%)±std) are reported

	IDIAP	ENTERFACE
Baseline Algorithm		
HMM	99.00 ± 0.61	68.20 ± 2.54
Fisher Score MultiClass Strategy		
B_{1vs1}	97.76 ± 0.87	65.92 ± 2.86
B_{1vs1R}	98.29 ± 1.01	66.84 ± 2.28
B_{1vsALL}	92.29 ± 2.72	58.64 ± 3.25
M_{FLC}	99.62 ± 0.49	NA
$\mathbf{M_{DLC}}$	$\mathbf{99.71 \pm 0.51}$	$\mathbf{72.98 \pm 1.25}$

6.6.2. Feature Selection and the Effect of HMM Parameters on the Classification Performance

Although one can use classifiers that do not suffer from the curse of dimensionality (such as SVMs), the dimensionality of the Fisher Score Space can be extremely high depending on the number of parameters of the underlying generative model and the input dimensionality. In this section, we investigate the effect of each HMM parameter on the classification accuracy. The computational cost of Fisher Score calculation can be decreased by considering only the most important parameters.

With the HMM as the underlying generative model, the normalized Fisher likelihood score space is:

$$\phi_{O_t} = diag(\Sigma_S)^{-\frac{1}{2}} \left[\begin{array}{cccc} log\ p(O_t|\theta) & \nabla_{a_{ii}} & \nabla_{w_{im}} & \nabla_{\mu_{im}} \quad \nabla_{vec(\Sigma)_{im}} \end{array} \right]^T \tag{6.11}$$

Thus, the number of features in this new feature space is:

$$C_{log\ p(O_t|\theta)} + C_{a_{ii}} + C_{\mu_{im}} + C_{\Sigma_{im}} + C_{w_{im}} \tag{6.12}$$

where

$$\begin{aligned} C_{log\ p(O_t|\theta)} &= 1 \\ C_{a_{ii}} &= N-1 \\ C_{\mu_{im}} &= NMV \\ C_{\Sigma_{im}} &= NMV^2 \\ C_{w_{im}} &= N(M-1) \end{aligned}$$

and N is the number of states, M is the number of mixtures and V is the number of input dimensionality.

Among the parameters of the HMM, the discriminative power of all parameter combinations are explored and the results are given in Table 6.4. The multi-class classifications are performed via M_{DLC} strategy.

All combinations of feature sets are explored and the best result is obtained by (μ, Σ) on the IDIAP dataset and (a, μ) on the eNTERFACE dataset. The results of these reduced sets are either equal or better than using the complete parameter set. If the parameters are used alone, the results show that most discriminatory features are the derivatives of the component means and variances. The least discriminatory feature is found to be the derivatives of the transition probabilities. The log-likelihoods on which the HMM decision is based are found to be less discriminative than expected. This result follows from the fact that each Fisher score space is processed independently with no regard to the relationship between log-likelihoods of the HMMs of each class. This relationship is apparently lost in the normalization process of the score spaces since the normalization of each score space is independent of others.

6.6.3. Dimensionality Reduction of Fisher Scores

We can reduce the dimensionality of the new feature space, by applying state-of-the art dimensionality reduction techniques. We compare two techniques, PCA and

Table 6.4. Effect of HMM parameters on the recognition performance. The abbreviations refer to the score spaces: ll for $logp(O_t|\theta)$, a for $\nabla_{a_{ii}}$, μ for $\nabla_{\mu_{ik}}$, and Σ for $\nabla_{\Sigma_{ik}}$

Selected HMM Parameters	IDIAP Test Acc. (%) & Std	eNTERFACE Test Acc. (%) & Std
(ll, a, μ, Σ)	**99.71 ± 0.51**	72.98 ± 1.25
(ll, a, μ)	99.33 ± 0.46	**73.99 ± 1.36**
(ll, a, Σ)	99.48 ± 0.57	73.07 ± 0.86
(ll, μ, Σ)	99.67 ± 0.45	72.98 ± 0.97
(a, μ, Σ)	99.67 ± 0.55	72.98 ± 0.93
$ll, a)$	81.86 ± 3.42	59.96 ± 3.92
(ll, μ)	99.33 ± 0.40	73.60 ± 1.29
(ll, Σ)	99.57 ± 0.61	72.94 ± 1.19
(a, μ)	98.95 ± 0.63	**73.99 ± 1.28**
(a, Σ)	99.62 ± 0.44	73.20 ± 0.96
(μ, Σ)	**99.71 ± 0.46**	72.94 ± 0.97
ll	76.81 ± 2.63	53.68 ± 1.55
a	45.62 ± 11.73	50.61 ± 4.11
μ	99.00 ± 0.52	73.51 ± 0.88
Σ	99.62 ± 0.49	72.98 ± 1.06

LDA, where the former aims to maximize the variance in the features and the latter aims to maximize the class separability in the new feature space. For PCA, we select the components that explain 99% of the variance. To apply LDA, we used the reduced feature space found by PCA.

The results are given in Table 6.5. In both datasets, accuracy obtained by PCA is less than the LDA accuracy, in the eNTERFACE06 datasets the difference is higher. In that dataset, the LDA accuracy is even higher than the baseline accuracy of the original Fisher mapping, without any dimensionality reduction.

Table 6.5. Dimensionality reduction

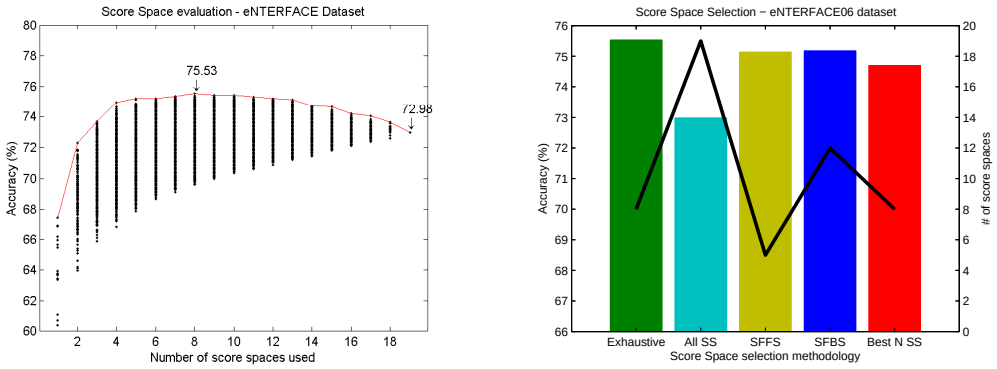
	IDIAP	ENTERFACE
No reduction	99.71 ± 0.51	72.98 ± 1.25
PCA	99.52 ± 0.59	58.07 ± 2.40
LDA	99.67 ± 0.45	74.17 ± 1.39

6.6.4. Score Space Selection

In the M_{DLC} strategy, each single classifier is capable of making a multi-class decision and their decisions are combined at the decision level to obtain an improved accuracy. Experiments show that sometimes a small subset of classifiers, or even a single classifier, may perform equally well (see Table 6.2). Hence our aim is to find techniques to select a subset out of K Fisher score mappings and use only the classifiers based on this subset.

We first run an exhaustive search for all the possible combinations of the score spaces. Figure 6.5 shows the results on the eNTERFACE dataset. With the 19 classes in the eNTERFACE dataset, one can have $2^{19} - 1$ possible subsets. The highest accuracy is obtained by using only eight score spaces.

Since exhaustive search is impractical for high number of classes, we implemented Sequential Floating Forward and Backward Search (SFFS, SFBS) strategies and also selecting the best of N score spaces (Best N SS). The results are given in Table 6.6. Although the accuracy of the subsets found by SFFS and SFBS are not as high as that of the exhaustive search, it is still higher than the accuracy of all score spaces. The result found by SFFS uses only five score spaces, with an accuracy of 75.13%. If we select best of N score spaces, the highest accuracy, 74.69%, is obtained with $N = 8$.



(a) (b)

Figure 6.5. Score space selection performances on eNTERFACE dataset (a) Exhaustive search over all score space combinations, (b) Score space selection strategies

Table 6.6. Score space selection results on eNTERFACE dataset

Method	% Accuracy	# of SS
Exhaustive	75.53	8
All SS	72.98	19
Best N SS	74.69	8
SFFS	75.13	5
SFBS	75.18	12

6.7. Conclusions

HMMs provide a good framework for recognizing hand gestures, by handling translation and scale variances and by modeling and processing variable length sequence data. However, performance can be increased by combining HMMs with discriminative models which are more powerful in classification problems. Fisher kernels are suitable for combining generative models with discriminative classifiers and theoretically, the resulting combined classifier has the powers of both approaches and has a better classification accuracy. However, as Fisher kernels are intrinsically binary, for multi-class classification problems such as gesture and sign recognition, the multi-class strategy must be defined properly in order to achieve high recognition accuracies.

In this study, we applied Fisher kernels to gesture and sign recognition problems and we proposed a multi-class classification strategy for Fisher Scores. The main idea of our multi-class classification strategy is to use the Fisher score mapping of one model in the classification process for all of the classes. As a result, each mapping is able to discriminate all the classes up to some degree. When all of these mappings are combined, higher accuracies are obtained when compared to the existing multi-class classification approaches in the literature.

7. Applications

In this thesis two applications have been designed and developed. The applications, SignTutor [2] and Signiary [113], are developed during the eNTERFACE'06 and eNTERFACE'07 workshops respectivey. The development of these applications have been through joint work, and will be acknowledged in the appropriate sections.

7.1. SignTutor: An Interactive System for Sign Language Tutoring

Sign Language, the natural communication medium for a deaf person, is difficult to learn for the general population. The prospective signer should learn specific hand gestures in coordination with head motion, facial expression and body posture. Since language learning can only advance with continuous practice and corrective feedback, we have developed an interactive system, called SignTutor, which automatically evaluates users' signing and gives multimodal feedbacks to guide them to improve their signing. SignTutor allows users to practice instructed signs and to receive feedback on their performance. The system automatically evaluates sign instances by multimodal analysis of the hand and head gestures. The time and gestural variations among different articulations of the signs are mitigated by the use of hidden Markov models. The multimodal user feedback consists of a text-based information on the sign, and a synthesized version of the sign on an avatar as a visual feedback. We have observed that the system has a very satisfactory performance, especially in the signer-dependent mode, and that the user experience is very positive.

7.1.1. SLR Assisted Sign Language Education

Practice can significantly enhance the learning of a language when there is validation and feedback. This is true for both spoken and sign languages. For spoken languages, students can evaluate their own pronunciation and improve to some extent by listening to themselves. Similarly, sign language teachers suggest that their students practice in front of the mirror. With an automatic SLR system, students can practice

by themselves, validate and evaluate their signing. Such a system would be called such as SignTutor, which would be instrumental in assisting sign language education, especially for non-native signers.

SignTutor aims to teach the basics of the sign language interactively. The advantage of SignTutor is that it automatically evaluates the student's signing and enables auto-evaluation via visual feedback and information about the goodness of the performed sign. The interactive platform of SignTutor enables the users to watch and learn new signs, to practice and to validate their performance. SignTutor automatically evaluates the users' signing and communicates them the outcome in various feedback modalities: a text message, the recorded video of the user, the video of the segmented hands and/or an animation on an avatar.

One of the key factors of SignTutor is that it integrates hand motion and shape analysis together with head motion analysis to recognize signs that include both hand gestures and head movements. This is very important since head movements are one aspect of sign language that most students find hard to perform in synchrony with hand gestures. To put this advantage of SignTutor, we have dwelled mostly on signs that have similar hand gestures and that are mainly differentiated by head movements. In view of the prospective users and the usage environment of SignTutor, we have opted for a vision-based user-friendly system which can work with easy to obtain equipment, such as webcams. The system operates with a single camera focused to the upper body of the user.

Figure 7.1 shows the graphical user interface of Sign Tutor. The system follows three steps for teaching a new sign: Training, practice and feedback. For the training phase, SignTutor provides a pre-recorded reference video for each sign. The users select a sign from the list of possible signs and watch the pre-recorded instruction video of that sign until they are ready to practice. In the practice phase, users are asked to perform the selected sign and their performance is recorded by the webcam. For an accurate analysis, users are expected to face the camera, with full upper body in the camera field of view, and wear a colored glove as marker. The recording continues until

Figure 7.1. SignTutor GUI: training, practice, information, synthesis panels and feedback examples

both hands are out of the camera. SignTutor analyzes the hand motion, hand shape and head motion in the recorded video and compares it with the selected sign.

We have integrated several modalities to the system for giving feedback to the user as for the quality of the enacted sign. The goodness criteria are given separately for the two components: the manual component (hand motion, shape, and position) and the non-manual component, (head and facial feature motion), together with the sign name with which it is confused (see Figure 1). Users can watch the video of their original performance. If the sign is properly performed, users may watch a caricaturized version of their performance on an animated avatar. A demonstration video of SignTutor can be downloaded from [114].

In summary, SignTutor aims to facilitate sign language learning especially for non-native beginners, by providing an interactive system. To assess the usability of

the overall system, we have performed a user study, with the students of the Turkish Sign Language beginner level course given in Boğaziçi University. We have collected the test scores and the comments of users to gauge its effectiveness.

7.1.2. SignTutor Modules

The block diagram of the SignTutor consists a face and hand detector stage, followed by the analysis stage, and the final sign classification box as illustrated in Figure 7.2. The critical part of SignTutor is the analysis and recognition sub-system which receives the camera input, detects and tracks the hand, extracts features and classifies the sign. In the sequel, we present the analysis and recognition modules and describe the synthesis and animation subsystem, which aims to provide a simple visual feedback environment for the users.

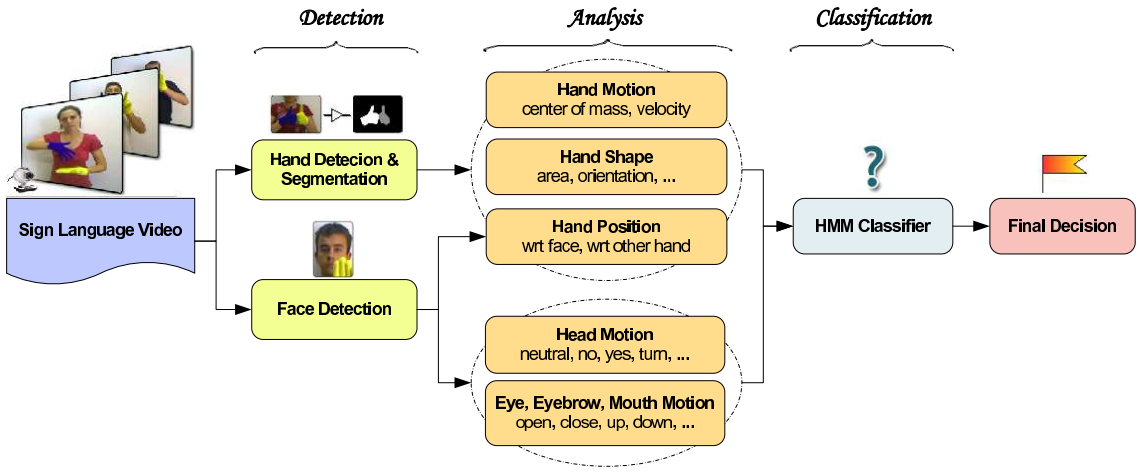


Figure 7.2. SignTutor system flow. Detection, analysis and classification steps

7.1.2.1. Hand Detection and Segmentation. Although skin color features can be applied for hand segmentation in controlled illumination, segmentation becomes problematic when skin regions overlap and occlude each other. In sign language, hand positions are often near the face and sometimes have to be in front of the face. Hand detection, segmentation and occlusion problems are simplified when users wear colored gloves. The use of a simple marker as a colored glove makes the system robust to changing

background and illumination conditions.

For each glove color, we train its respective histogram of color components using several images. We have chosen HSV color space due its robustness to changing illumination conditions [115]. The H, S and V components are quantized into bins. At each bin of the histogram, we calculate the number of occurrences of pixels that correspond to that bin, and finally the histogram is normalized. To detect hand pixels in a scene, we find the histogram bin it belongs to and apply thresholding. We apply double thresholding, set at low and high values, to ensure connectivity: A pixel is considered as a hand pixel if its histogram value is higher than the high threshold. If it is between the two thresholds it is still labeled as glove (hand), provided one or more of neighbor pixels were labeled as glove. The final hand region is assumed to be the largest connected component over the detected pixels.

7.1.2.2. Analysis of Hand Motion. The system processes the motion of each hand by tracking its center of mass (CoM) and estimating in every frame the position and velocity of each segmented hand. However, the hand trajectories can be corrupted by segmentation noise. Moreover, hands may occlude each other or there may be sudden changes in lighting (e.g., a room light turned on or off), which may result in detection and segmentation failures. Thus, we use two independent Kalman filters, one for each hand, to smooth the estimated trajectories. The motion of each hand is approximated by a constant velocity motion model, hence acceleration is neglected. When the system detects a hand in the video, it initializes the corresponding Kalman filter. Before each sequential frame, Kalman filter predicts the new hand position, and the filter parameters are updated with the hand position measurements found by the hand segmentation step. We calculate the hand motion features from the posterior states of the corresponding Kalman filter: x, y coordinates of CoM and velocity [8]. When there is a detection failure due to occlusion or bad lighting, we only use the Kalman filter prediction without updating the filter parameters. Finally, the system assumes that the hand is out of the camera view if no hand segment can be detected for a number of consecutive frames.

The trajectories must be further normalized to obtain translation and scale invariance. We use a normalization strategy similar to [8]. The normalized trajectory coordinates are calculated via min-max normalization. The translation normalization is handled by calculation of the mid points of the range of x and y coordinates, denoted as x_m, y_m. The scaling factor, d, is selected to be the maximum of the spread in x and y coordinates, since scaling horizontal and vertical displacements with different factors disturbs the shape. The normalized trajectory coordinates, $(< x_1; y_1 >; ...; < x_t; y_t >; ...; < x_N; y_N >)$ such that $0 \leq x_t, y_t \leq 1$, are then calculated as follows:

$$x'_t = 0.5 + 0.5(x_t - x_m)/d \tag{7.1}$$

$$y'_t = 0.5 + 0.5(y_t - y_m)/d \tag{7.2}$$

Since signs can also be two handed, both hand trajectories must be normalized. However, normalizing the trajectory of the two hands independently may result in a possible loss of data. To solve this problem, the midpoint and the scaling factor of the left and right hand trajectories are calculated jointly. Following the trajectory normalization, the left and right hand trajectories are translated such that their starting position is at $(0, 0)$.

7.1.2.3. Extracting Features from a 2D Hand Shape. Hand shape and finger configuration contribute significantly to sign identification, since each sign has a specific movement of the head, hands and hand postures. Moreover, there are signs which solely depend on the hand shape. Our system is intended to work with a single low-resolution camera whose field of view covers the upper body of the user, hence is not directly focused on the hands. In this setup, we face several difficulties:

- Low resolution of hand images, where each hand image is smaller than 80x80 pixels,
- Segmentation errors due to blurring caused by fast movement

- More than one hand posture can result in the same binary image (silhouette).

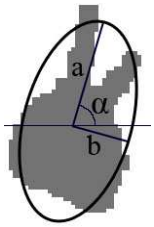
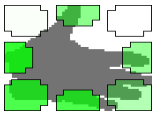
These problems constrain us to use only low-level features, which are robust to segmentation errors and work well with low resolution images. Therefore we use simple appearance-based shape features calculated from the hand silhouettes. The features are selected to reflect differences in hand shape and finger postures. They are also required to be scale invariant so that hands with similar shape but different size result in the same feature values. However recall that our system uses a single camera, hence we do not have depth information, except for the foreshortening due to perspective. In order to maintain some information about the z-coordinate (depth), five of the 19 features were not scale normalized. Prior to the calculation of the hand shape features, we take the mirror reflection of the right hand so that we analyze both hands in the same geometry; with thumb to the right. All 19 features are listed in 7.1. It is important to note that these features are not invariant to viewpoint, and the users are required to sign facing the camera for an accurate analysis. The classifier is tolerant of small rotations that can naturally occur while signing.

Seven of the features (#1,2,4,5,6,7,8) are based on using the best fitting ellipse (in the least-squares sense) to the hand silhouette. The inclination angle assumes values in the range $[0, 360]$. In order to represent the 4-fold symmetry of the ellipse, we use $\sin(2\alpha)$ and $\cos(2\alpha)$ as features, where α is in the range $[0, 180]$.

Features #9 to 16 are based on using "area filters". The bounding box of the hand is divided into eight areas, in which, percentage of hand pixels are calculated. Other features in 7.1 are perimeter, area and bounding box width and height.

All 19 hand shape features are normalized into values between 0 and 1. This is obtained by dividing the percentage features (#9 to 16) by 100, and the cardinal number features by their range, that is, by using $F_n = (F - min)/(max - min)$ where min is minimum value of the feature in the training dataset and max is maximum value. Any value exceeding the $[0, 1]$ interval is truncated.

Table 7.1. Hand shape features

#		Feature	Invariant	
			Scale	Rotation
1		Best fitting ellipse width		✓
2		Best fitting ellipse height		✓
3		Compactness (perimeter^2/area)	✓	✓
4		Ratio of hand pixels outside / inside of ellipse	✓	✓
5		Ratio of hand / background pixels inside of ellipse	✓	✓
6		$sin(2\alpha)$ α = angle of ellipse major axis	✓	
7		$cos(2\alpha)$ α = angle of ellipse major axis	✓	
8		Elongation (ratio of ellipse major/minor axis length)	✓	✓
9		Percentage of NW (north-west) area filled by hand	✓	
10		Percentage of N area filled by hand	✓	
11		Percentage of NE area filled by hand	✓	
12		Percentage of E area filled by hand	✓	
13		Percentage of SE area filled by hand	✓	
14		Percentage of S area filled by hand	✓	
15		Percentage of SW area filled by hand	✓	
16		Percentage of W area filled by hand	✓	
17		Total area (pixels)		✓
18		Bounding box width		
19		Bounding box height		

7.1.2.4. Analysis of Head Movements. Once the face is detected, rigid head motions such as head rotations and head nods are determined by using an algorithm inspired by the human visual system. First, we apply a filter following the model of the human retina [116]. This filter enhances moving contours with Outer Plexiform Layer (OPL) and cancels static ones with Inner Plexiform Layer (IPL). This prefiltering mitigates any illumination changes and noise. Second, we compute the fast Fourier transform (FFT) of the filtered image in the log polar domain as a model of the primary visual cortex (V1) [99]. This step allows extracting two types of features: the quantity of motion and motion event alerts. In parallel, an optic flow algorithm extracts both orientation and velocity information only on the motion event alerts issued by the visual cortex stage [117]. Thus, after each motion alert, we estimate the head velocity at each frame. After these three steps, the head analyzer outputs three features per frame: the quantity of motion, and the vertical and horizontal velocities.

These three features provided by the head motion analyzer can vary with different

performances of the same sign. Moreover, head motion is not directly synchronized with the hand motion. To handle the inter- and intra-subject differences, weighted average smoothing is applied to head motion features, with $\alpha = 0.5$. The smoothed head motion feature vector at time i, F_i, is calculated as $F_i = \alpha\ F_i + (1 - \alpha)\ F_{i-1}$. This smoothing has the effect of mitigating the noise between different performances of a sign and creating a slightly smoother pattern.

7.1.2.5. Preprocessing of sign sequences. The video sequences obtained contain frames where the signer is not performing any sign (beginning and terminating parts) and some frames that can be considered as transition frames. These frames of the sequence are eliminated by considering the result of the hand segmentation step:

- All frames at the beginning of the sequence are eliminated until a hand is detected.
- If the hand fails to be detected during the sequence for less than N consequent frames, the shape information is copied from the last frame where there was still detection to the current frame.
- If the hand fails to be detected for more than N consequent frames, the sign is assumed to be finished. The remaining frames including the last N frames are eliminated.
- After these prunings, transition frames, defined as the T frames from the start and end of the sequence, are deleted.

7.1.2.6. Classification: A Sequential Fusion Approach. At the classification phase, we have used HMMs to model each sign and the classification decision is given via the likelihood of the observation sequence with respect to each HMM. HMMs are preferred in gesture and sign recognition as the changes in the speed of the performed sign or slight changes in the spatial domain are successfully handled.

The classification module receives all the features calculated in the hand and head analysis modules as input. For each hand, there are four hand motion features (position and velocity in vertical and horizontal coordinates), 19 hand shape features

and three head motion features per frame. We also use the relative position of the hands with respect to the face CoM, yielding two additional features (distance in vertical and horizontal coordinates) per hand per frame. We normalize these latter distances by the face height and width, respectively, and use the normalized and pre-processed sequences to train the HMM models. Each HMM model is a continuous 4-state left-to-right model and is trained for each sign, using the Baum-Welch algorithm.

We use the sequential likelihood fusion method for combining manual and non-manual parts of the sign, as explained in Section 5.4.4.1. The strategy uses the fact that there may be similar signs which differ slightly and cannot be classified accurately in an "all signs" classifier. These signs form a cluster and their intra-cluster classification must be handled in a special way. Thus, our sequential fusion method is based on two successive classification steps: In the first step, we perform an inter-cluster classification and in the second step we do intra-cluster classifications. Since we want our system to be as general as possible and adaptable to new signs, we do not use any prior knowledge about the sign clusters. We let the system discover potential sign clusters, that are similar in manual gestures, but that differ in non-manual motions. Instead of rule-based programming of these signs, we opt to extract the cluster information as a part of the recognition system.

7.1.2.7. Visual Feedback via a Synthesized Avatar. As one of the feedback modalities, SignTutor provides a simple animated avatar that mimics the users' performance for the selected sign. The synthesis is based on the features extracted in the analysis part. Currently the avatar only mimics the head motion of the user and hand motions are animated from a library.

Our head synthesis system is based on the MPEG-4 Facial Animation Standard [118]. As input, it receives the motion energy, the vertical and horizontal velocities of the head motion and the target sign as well. It then filters and normalizes these data in order to compute the head motion during the considered sequence. The result of the processing is expressed in terms of Facial Action Points (FAP) and is fed into

the animation player. For head animation, we use XFace [119], an MPEG-4 compliant 3D talking head animation player. The hands and arms synthesis system is based on OpenGL, and an animation is prepared explicitly for each sign. We merge the head animation with the hands and arms animation to form an avatar with full upper body.

7.1.3. Evaluation of the System Accuracy

7.1.3.1. Classification by Using Only Manual Information. This classification is done via HMMs that are trained only with the hand gesture information related to the signs. Since hands form the basis of the signs, these models are expected to be very powerful in classification. However, absence of the head motion information precludes correct classification when signs differ only in head motion. We denote these models as HMM_M.

7.1.3.2. Feature Fusion. The manual information and the non-manual information can be combined in a single feature vector to jointly model the head and hand information. Since there is not a direct synchronization between hand and head motions, these models are not expected to have much better performance than HMM_M. However using head information results in a slight increase in the performance. We denote these models as $\text{HMM}_{M\&N}$.

7.1.3.3. Sequential Fusion. The aim of this fusion approach is to apply a two-tier cluster-based sequential fusion strategy. The first step identifies the cluster of the performed sign within a general model, $\text{HMM}_{M\&N}$, and the confusion inside the cluster is resolved at the second step, with a dedicated model, HMM_N, which uses only head information. The head motion is complementary to the sign thus it cannot be used alone to classify the signs. However, in the sequential fusion methodology, indicated as $\text{HMM}_{M\&N}$ HMM_N, they are used to perform intra-cluster classification. We report the results of our sequential fusion approach with different clusters, first on base sign clusters and then on automatically identified clusters based on the joint confusion matrix.

Table 7.2. Signer-Independent test results

	Sbj1	Sbj2	Sbj3	Sbj4	Sbj5	Sbj6	Sbj7	Sbj8	Average
					Base Sign Accuracy				
$\mathbf{HMM}_M$	100	100	100	100	96.84	100	100	100	99.61
$\mathbf{HMM}_{M\&N}$	100	100	100	100	97.89	100	100	100	99.74
$\mathbf{HMM}_{M\&N} \Rightarrow \mathbf{HMM}_N$	100	100	100	100	97.89	100	100	100	99.74
					Overall Accuracy				
$\mathbf{HMM}_M$	66.32	77.9	60.00	71.58	57.9	81.05	52.63	70.53	67.24
$\mathbf{HMM}_{M\&N}$	73.68	91.58	71.58	81.05	62.11	81.05	65.26	77.89	75.53
$\mathbf{HMM}_{M\&N} \Rightarrow \mathbf{HMM}_N$	82.11	72.63	73.68	88.42	56.84	80.00	81.05	71.58	75.79
Base Sign Clusters									
$\mathbf{HMM}_{M\&N} \Rightarrow \mathbf{HMM}_N$	85.26	76.84	77.89	89.47	63.16	80.00	88.42	75.79	**79.61**
Auto. Identified Clusters									

We report two sets of results: the base sign accuracy and the overall accuracy. To report the base sign accuracy, we assumed that a classification decision is correct if the classified sign and the correct sign are in the same base sign cluster. The base sign accuracy is important for the success of our sequential fusion method based on sign clusters. The overall accuracy reports the actual accuracy of the classifier over all the signs in the dataset. These accuracies are reported on the two protocols: the signer-independent protocol and the signer-dependent protocol.

Signer-Independent Protocol. In the signer-independent protocol, we constitute the test sets from instances of a group of subjects in the dataset and train the system with sample signs from the rest of the signers. For this purpose, we apply an 8-fold cross-validation, where at each fold test set consists of instances from one of the signers and the training set consists of instances from the other signers. In each fold there are 665 training instances and 95 test instances. The results of signer-independent protocol are given in Table 7.2

The base sign accuracies of each of the three classifiers in each fold are 100% except for the fifth signer, which is slightly lower. This performance result shows us

that a sequential classification strategy is appropriate, where specialized models are used in a second step to handle any intra-cluster classification problem.

The need for the usage of the head features can be deduced from the high increase of the overall accuracy with the contributions of non-manual features. With $\text{HMM}_{M\&N}$, the accuracy increases to 75.5% as compared to the accuracy of HMM_M, 67.2%. Further increase is obtained by using our sequential fusion methodology with automatically defined clusters. We also report the accuracy of the sequential fusion with the base sign clusters, to show that using those semantic clusters results in a 4% lower accuracy than automatic clustering.

Signer-Dependent Protocol. In the signer-dependent protocol, we put examples from each subject in both of the test and training sets, although they never share the same sign instantiation. For this purpose, we apply a 5-fold cross validation where at each fold and for each sign, four out of the five repetitions of each subject are placed into the training set and the remaining one to the test set. In each fold there are 608 training examples and 152 test examples. The results of signer dependent protocol are given in Table 7.3.

The base sign accuracies of each of the three classifiers are similar to the signer-independent results. The overall accuracies become much higher and the sequential fusion technique does not contribute significantly this time. This is probably a result of the ceiling effect and the differences between the approaches are not apparent as a result of the already high accuracies.

7.1.4. User Study

We have conducted a user study to measure the real-life performance of SignTutor and to assess the usability of the overall system. Our subjects were volunteers, two males and four females, six out of seven students taking an introductory Turkish Sign Language course given in Bogazici University. Two of the students (one male and one

Table 7.3. Signer-dependent test results

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
			Base Sign Accuracy			
$\mathbf{HMM}_M$	99.34	100	98.68	100	100	99.61
$\mathbf{HMM}_{M\&N}$	100	100	98.68	100	100	99.74
$\mathbf{HMM}_{M\&N} \Rightarrow \mathbf{HMM}_N$	100	100	98.68	100	100	99.74
			Overall Accuracy			
$\mathbf{HMM}_M$	92.76	89.47	89.47	94.08	92.76	91.71
$\mathbf{HMM}_{M\&N}$	92.76	95.39	93.42	96.05	94.08	94.34
$\mathbf{HMM}_{M\&N} \Rightarrow \mathbf{HMM}_N$	92.76	95.39	92.76	95.39	94.08	94.08
Base Sign Clusters						
$\mathbf{HMM}_{M\&N} \Rightarrow \mathbf{HMM}_N$	92.76	95.39	92.76	96.05	94.08	94.21
Auto. Identified Clusters						

female) are from the Computer Engineering Department and the rest are from the Foreign Language Education department. All subjects were highly motivated for the experiment and were excited about the SignTutor when they were first told about the properties of the system.

We performed the experiments in two sessions, where in each session, subjects were asked to practice three signs. The second session was conducted with a time lapse of at least one week after the first session. Before the first session, we present the SignTutor interface to the subjects in the form of a small demonstration. At the second session, the users are expected to start using the SignTutor without any presentation. For each subject, we measured the time on task, where each task is defined as the learning, practicing and evaluating one of the three signs. At the end of the experiment, we interviewed the subjects and asked them to fill a small questionnaire.

We asked three questions in the questionnaire and the subjects scored at five levels, from strongly disagree (1) to strongly agree (5). The results are shown in Figure 7.3. The average scores for all questions are above four, indicating the favorable views of the subjects on the usability of the system.

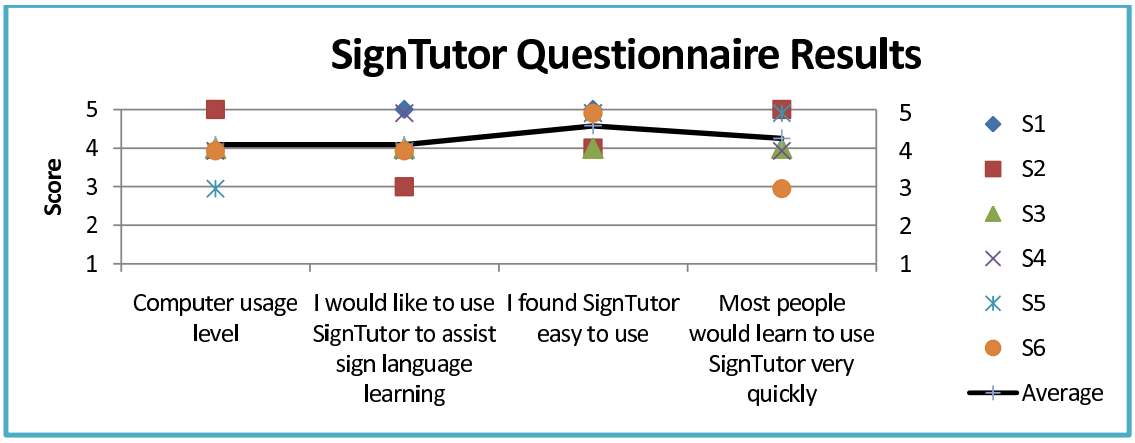
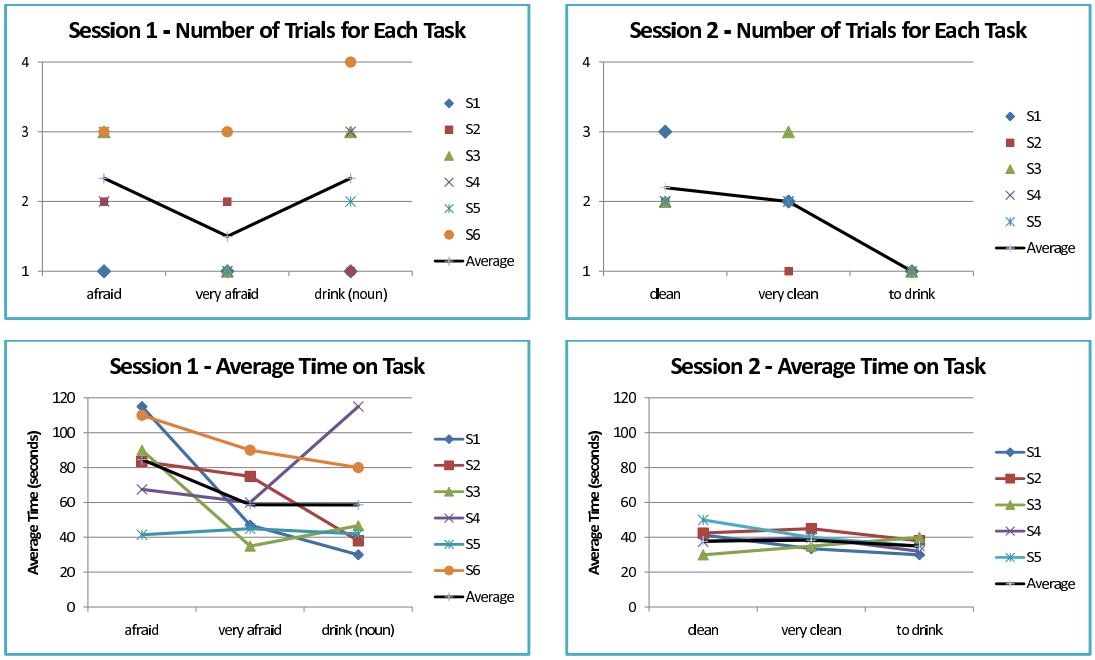


Figure 7.3. Usability questionnaire results

In session 1, the subjects are asked to practice three signs: AFRAID, VERY AFRAID and DRINK (NOUN). These three signs are selected such that the first two are variants of the same base sign, performed by two hands; and the third is a completely different sign, performed by a single hand. We measured the number of seconds for each task and the results are shown in Figure 7.4. The subjects practiced each sign until they received a positive feedback. The average number of trials is around two. The average time is 85 seconds for the first task and decreases to 60 seconds for the second and third tasks. These results show that after the first task, the subjects are more comfortable in using the system.

In session 2, the subjects are asked to practice another set of signs: CLEAN, VERY CLEAN and TO DRINK. As in the first session, the first two are variants of the same base sign, performed by two hands; and the third is a completely different sign, performed by a single hand. From the six subjects who participated in the first session, only five participated in the second one. In this session, the subjects directly started using the system without any help about the usage. The results are shown in Figure 5 and they reflect that the subjects recall the system usage, without any difficulty, and the average time on task has decreased with respect to the first session. The average number of trials is again around two for the first two signs, similar to the first session, which shows that the subjects perform a new sign correctly after two trials on average. For the third sign, all the subjects succeeded in their first trial.



(a) (b)

Figure 7.4. Task analysis for (a) Session 1 and (b) Session 2. For each session, the number of trials and the average time on task is plotted

During the interviews, the subjects made positive comments about the system in general. They find the interface user friendly and the system easy to use. All of the subjects indicate that using SignTutor during sign language learning can help to learn and recall the signs. The subjects find the practice part, especially its capability to analyze both hand and head gestures, very important. They commented that the possibility of watching the training videos and their own performance together with the animation makes it easier to understand and perform the signs. They note, however, that it would be nice to have more constructive feedback about their own performance, explaining what is wrong and what should be corrected. One of the subjects found the decisions of the SignTutor very sensitive and she noted that the decision-making can be relaxed.

The subjects had no difficulty in using the glove but they commented that it would be better to be able to use the system without any gloves. One of the subjects suggested adding a picture, possibly a 3D image, of the hand shape used during the sign in addition to the sign video. This will help to understand the exact hand shape

performed during the signing. During the experiments, we were able to observe the system performance in a real life setting. A very important requirement of the system is that the camera view should contain the upper body of the user and the distance between the camera and the user should be at least one meter. We have performed the experiments in an office environment with normal lighting conditions, without using any special lighting device. The hand segmentation requires a reasonably illuminated office environment and the user's clothes should have a different color than the gloves. If these conditions are met, the system works accurately and efficiently.

7.1.5. Acknowledgments

SignTutor is developed as a group project during the eNTERFACE 2006 Workshop on Multimodal Interfaces [2]. I would like to acknowledge the other members of the group and their work on the development of SignTutor: graphical user interface is implemented by Ismail Ari, head motion feature extraction is developed by Alexandre Benoit, hand shape feature extraction is developed by Pavel Campr and sign synthesis is implemented by Ana Huerta Carrillo, Franois-Xavier Fanard.

7.2. Signiary: An Automatic Turkish Sign Dictionary

We present, Signiary, an automatically created sign dictionary where the user enters a word as text and retrieves videos of the related sign. The word is searched from a collection of videos recorded from the broadcast news for the hearing impaired. In these videos, the news is presented by multiple modalities: the speaker also signs with the hands as she talks and there is corresponding sliding text. We retrieve the occurrences of the entered word from the news videos, via the recognized speech and sliding text. The retrieved videos are further analyzed to detect clusters among the signs that reflect pronunciation differences or sign homonyms.

As the video source, we used videos recorded from the Turkish broadcast news for the hearing impaired. The news video consists of three major information sources: sliding text, speech and signs. Figure 7.5 shows an example frame from the recordings.

The three sources in the video convey the same information via different modalities. The news presenter signs the words as she talks. However, sign languages have their own grammars and word orderings and it is not necessary to have the same word ordering in a Turkish spoken sentence and in a Turkish sign sentence [59]. Thus, the signing in these news videos should not be considered as Turkish sign language (Turk Isaret Dili, TID) but Signed Turkish: the sign of each word is from TID but their ordering would have been different in a proper TID sentence. In addition to the speech and sign information, a corresponding sliding text is superimposed on the video. Our methodology is to process the video to extract the information content in the sliding text and speech components to generate segmented and annotated sign videos. The main goal is to use this annotation to form a sign dictionary. Once the annotation is completed, techniques are employed to check pronunciation differences or homonyms among the retrieved signs.

Figure 7.5. An example frame from the news recordings. The three information sources are the speech, sliding text, signs

7.2.1. System Information

The application receives the text input of the user and attempts to find the word in the news videos by using the other modalities that convey the same information.

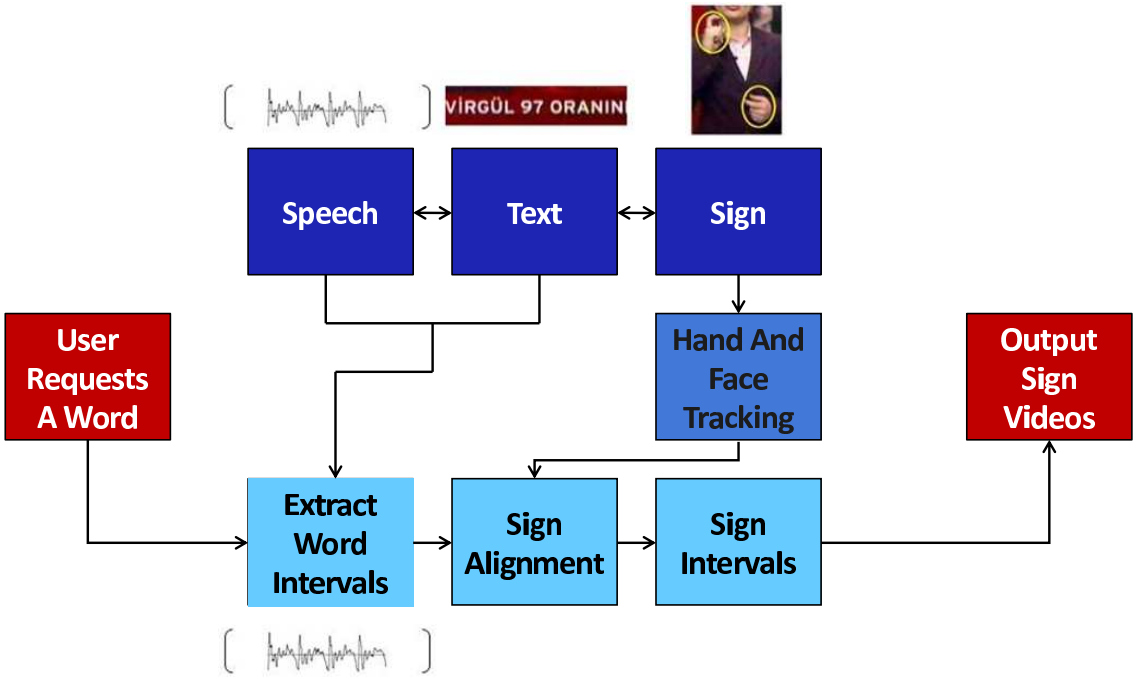


Figure 7.6. Modalities and the system flow

By using a speech recognizer, the application returns several intervals from different videos that contain the entered word. If the resolution is high enough to analyze the lip movements, audio-visual analysis can be applied to increase accuracy. Then, sliding text information may optionally be used to control and correct the result of the retrieval. This is done by searching for the word in the sliding text modality during each retrieved interval. If the word can also be retrieved by the sliding text modality, the interval is assumed to be correct. The sign intervals are extracted by analyzing the correlation of the signs with the speech. Sign clustering is necessary for two reasons. First, there can be false alarms of the retrieval corresponding to some unrelated signs and second, there are homophonic words that have the same phonology but different meanings; thus possibly different signs. The system flow is illustrated in Figure 7.6.

7.2.2. Spoken Term Detection

Spoken Term Detection (STD) is used as a tool to retrieve the signs in the news videos based on speech information. An HMM based speech-to-text system converts the audio data into a textual representation. When the user enters a query, the retrieval

engine returns program name, time and relevance information of the corresponding query [120].

7.2.3. Sliding Text Recognition

The news videos are accompanied by a sliding text band, which includes simultaneous transcription of what is being said. It is placed at the bottom of the screen and contains characters with a specific font, displayed in white pixels over a solid background. Speed of the sliding text is approximately constant throughout the whole video sequence (4 pixels/frame), which allows each character to appear on the screen for at least 2.5 seconds.

Sliding Text Recognition (STR) is used to control and correct the result of the retrieval. Sliding text band is found at the first frame, by calculating horizontal and vertical projection histograms of the binarized image. Then each character is cropped from the figure and recognized by the template matching method using Jaccard's distance [113].

7.2.4. Sign Analysis

7.2.4.1. Hand and Face Tracking. We apply a joint particle filter based on the joint likelihood calculation for the hands and the face, which ensures that the particles of each object will not jump to other objects and the tracking continues without problems during and after occlusion (see Chapter 4).

7.2.4.2. Sign Alignment and Clustering. The tracking algorithm provides five features per object of interest, 15 features in total. Gaussian smoothing of measured data in time is applied to reduce noise. We then translate the origin to the head center and scale with respect to the width of the head ellipse. In total, 30 features from tracking are provided, 15 smoothed features obtained by the tracking algorithm and 15 differences.

After obtaining relevant sequences using the retrieval engine, we use Dynamic Time Warping (DTW) algorithm for alignment. We use the coordinates and the coordinate differences between two consecutive center points of the hands. We align sequences pairwise using DTW and find the best aligning paths between each pair. Then we combine these paths using a progressive multiple alignment algorithm and find the common parts that minimize the alignment scores for each sequence.

After the alignment, we want to find out whether the signs in those videos are the same or are different (in case of homonyms or mistakes of the speech recognizer). For this purpose we use clustering. The goal is to calculate the similarity of these two signs and to determine if they are two different performances of the same sign or two totally different signs. If the signs are considered to be the same, they should be added to the same cluster.

To cluster multiple signs, we construct a dendrogram tree and determine the separation level:

1. Calculate pairwise distances between all compared signs and store those distances in an upper triangular matrix
2. Group two most similar signs together and recalculate the distance matrix (it will have one less row and column)
3. Repeat step 2. until all signs are in one of the groups.
4. Mark the highest difference between two distances as the distance up to which the signs are in the same cluster

7.2.5. System Integration and Evaluation

The speech and sliding text modalities are integrated via a cascading strategy. Output of the STR is used as a verifier on the STD. STD hypotheses are checked with STR and accepted if the word can also be found in the sliding text modality. The retrieved intervals are given to the sign analysis module and the sign intervals are extracted as a result of the sign alignment step.

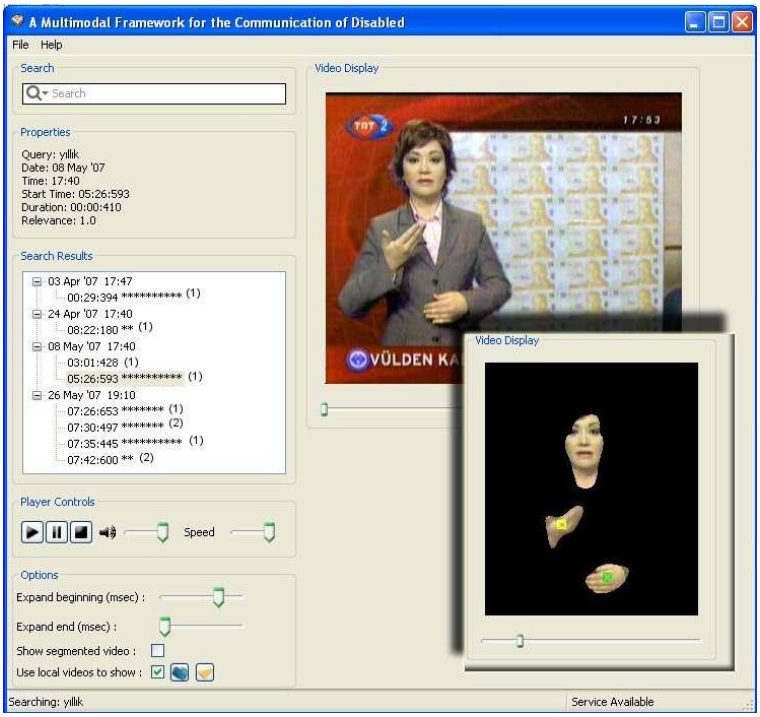


Figure 7.7. Screenshot of the user interface

We have tested the overall system performance on a manually annotated ground truth data of 15 words. For each selected word, the system retrieves all 10 occurrences, except two words, which have eight occurrences. Among these 146 retrievals, 134 of them are annotated as correct, yielding 91.8% correct retrieval accuracy. If we extend the intervals by 0.5 seconds from the start and the end, in 100% of the correct retrievals, the sign is contained in the interval.

The screenshot of the user interface is shown in Figure 7.7. The "Search" section is where the user inputs a Turkish word or a phrase and sends the query. The application communicates with the engine on the server, retrieves data and processes it to show the results. The stars next to each result are used to inform the user about the reliability of the found results. When the user selects a result, it is played in the "Video Display" window. The original news video or the tracking results is shown in this section according to the user's selection. "Player Controls" and "Options enable the user to expand the duration to left/right or adjust the volume/speed of the video to analyze the sign in detail.

7.2.6. Acknowledgments

Signiary is developed as a group project during the eNTERFACE 2007 Workshop on Multimodal Interfaces [113]. I would like to acknowledge the other members of the group and their work: graphical user interface is implemented by Ismail Ari, STD is developed by Siddika Parlak, STR is developed by Erinc Dikici, skin color detection is developed by Marek Hruz. For sign alignment and clustering, we worked together with Pavel Campr and Pinar Santemiz.

8. Conclusions

The focus of this thesis is the advancement of the state of the art in vision based hand gesture and sign language recognition systems, in three main tasks: accurate markerless tracking in a less restricted environment, integration of manual and non-manual components and better classification models to increase the classification performance in isolated sign recognition. For each task, we developed principled techniques that will enable their use, not only in sign language or hand gesture recognition but also in other related areas of computer vision.

We aim to design and develop methods and systems that can work on inexpensive and easy to find hardware, such as ordinary computers with ordinary equipment. With this perspective, the vision based recognition methods presented in this thesis rely on commercial, easy-to-find web cameras. Although hand gestures and signs are performed in 3D and a stereo camera system can be able to extract 3D information as well, we used single camera setup in most of our study. The advantage of the single camera setup is that it can be easily obtained and used by every user, experienced or inexperienced, and single web cameras are now a standard part of the computer equipment, and are included in laptops as standard equipment.

The joint particle filter method proposed in this thesis provides a robust multiple object tracking approach for problems where there are multiple identical objects and the objects do not move independently, with frequent interactions as well as occlusions. In order to be able to achieve real time or close to real time tracking, instead of modeling the whole upper body and the arms, we model only the two hands and the face, and their respective motion and positions. For all the objects, a single joint likelihood is calculated, which takes into account the relative motion and position of the objects. We have performed experiments on normal speed signing videos, recorded from Turkish broadcast news for the hearing impaired, and achieved around 97% accuracy on a 15 minute recording. As the computational complexity of the particle filter depends on the number of particles, achieving real time accuracy is only possible by reducing the

number of particles. For this purpose, we have integrated the mean shift algorithm to the particle filter, which allows us to achieve similar accuracies with fewer number of particles. Our experiments show that for medium resolution images the proposed method runs close to real-time with high accuracy. The main disadvantage of the method is the excessive number of parameters that need to be set. Although some of the parameters can be automatically estimated, for most of the parameters, manual or semi automatic determination, based on the image resolution and the distance between the camera and the subject, is necessary.

The recognition of signs with manual and non-manual components is a challenging task and is rarely addressed in SLR literature. In this thesis, we present one of the frontier works on the subject and propose a sequential fusion strategy for the integration of manual signs and non-manual signals in a recognition framework. Non-manual signals in sign language are displayed in the form of head movements, facial expressions and body posture. In our study, we consider the head movements and partly, body motion. Although facial expressions are frequently used as non-manual signals, currently, accurate analysis of facial expressions is only possible with high resolution face areas. As we use a single camera setup, and record the whole upper body with medium resolution, the face resolution is low. Moreover, our aim in this study is to model the temporal characteristics and design a recognition framework that integrates the manual and non-manual components. As tracking and feature extraction of facial expressions is another challenging topic, the analysis of facial expressions in non-manual signals is not considered in this study and left as a future direction.

The eNTERFACE ASL database is collected during this thesis and is used to evaluate the performance of the methods developed. This database contains variations of base signs in the form of variations in manual signing and also in the form of non-manual signals. Moreover, we have collected the database from eight signers, which allow us to make experiments on signer independent recognition tasks. Although the number of signs in the database is small, the high number of signers makes it a good test bed to experiment new signer independent models and approaches.

Gesture and sign sequences are variable length sequences that contain both the spatial and temporal information about the performed hand gesture. As in speech recognition literature, HMMs have outperformed other approaches and become the state-of-the art in modeling gestures and signs with their power of modeling spatial and temporal information and their ability to work with variable length data. HMMs are generative models in which for each gesture, sign or a sub-unit, a dedicated HMM model should be trained. Standard HMMs are pure generative models and only positive examples are used to train the models.

Generative models provide a good framework, especially for difficult data types such as speech, vision, text and biosequence data, where sequences of array of variable sizes are processed. However, for classification problems, discriminative methods are superior as they provide decision boundaries. In order to have better classification accuracy when processing difficult data types, we should investigate models that combine the powers of generative and discriminative methods. Fisher kernels are an example of such methods.

In this thesis study, the combination of generative and discriminative approaches is facilitated via Fisher scores derived from HMMs. These Fisher scores are then used as the new feature space and trained using a SVM. We have shown that the performance of the combined classifier is directly related to the multi-class classification strategy and should be carefully selected. We propose a new multi-class classification scheme which outperforms the existing multi-class classification strategies. The main disadvantage of Fisher kernels is the increased computational complexity. We also present several methods to decrease the computational complexity, based on dimensionality reduction, score space selection, and feature selection, and show that the computational complexity can be reduced without affecting the performance. However, the methods to decrease the computational complexity and the suitability of this approach in a real-time setting should be further investigated.

Finally, we present two applications that are based on the methods and approaches presented in this thesis. The SignTutor application is designed such that it

can be used by a wide range of users in home or office environments, just by connecting a single web camera to the computer. The performed user study has shown that the users find the system easy to use and helpful in learning signs. The Signiary application provides an automatically created sign dictionary, where the users can search for signs and see different pronunciations, homonyms of the signs and their usage in signed Turkish. Both of these applications are at their first prototype phase and can be improved to increase the performance and usability and to decrease the computational complexity.

Sign language recognition is a challenging and relatively new research area. Just as spoken languages, sign languages have a grammar, morphology, syntax, co-articulation, accents, pronunciation differences and homonyms. However, linguistic studies on sign languages are still at the preliminary stage, and not all the properties of sign languages are yet revealed. Besides the linguistic challenges, the technical challenges of sign language recognition are related to several facts: First, the source of signs is a video which is computationally expensive to process. Second, the environment, the clothing, the background and also the occlusions, camera view, affect the processing quality, resulting poor quality features. Third, sign languages use the visual world and use multiple visual modalities in parallel to convey the message. The analysis and integration of these multiple modalities is challenging both due to its computational complexity but also due to the difficulty of the task. This thesis study addresses these challenging problems in sign language recognition and proposes several techniques for the solution of these problems. Although our results are not conclusive, we obtain promising results for each of these challenging problems, which encourage further research.

The future directions on sign language recognition are on signer independent, large vocabulary systems in both isolated and continuous recognition tasks. All kinds of non-manual information, including head motion, body posture and facial expressions, should be further analyzed and fully integrated with the manual signs. The most important issue in this integration task is on how to handle the different temporal characteristics of the manual and non-manual signals. Not only the integration of manual and non-manual components but also the integration of the components within

manual signs, including hand shape, motion and position information for both hands, is crucial to achieve higher classification accuracy. Variants of Bayesian networks, such as coupled HMMs, input-output HMMs, or other approaches can be designed and utilized. These models should be able to handle diverse sign types: signs that use both hands or a single hand, signs that have a global motion, a local motion or no motion at all, etc. Hand shape analysis should be further investigated for signs with global motion. Linguistic studies show that when there is fast, global motion, the hand shape is less important. Furthermore, during fast motion, the hand image can be blurred due to low camera frame rate and also the frontal view of the hand shape is changed and self-occluded due to rotations. To overcome these difficulties, event-based hand shape can be investigated, where the analysis is only done on specific events, in which the exact hand shape is visible and informative.

APPENDIX A: Hidden Markov Models

A.1. Definition

The elements of an HMM are prior probabilities of states, π_i, transition probabilities, a_{ij} and observation probabilities, $b_i(O_t)$ where $1 \leq i \leq N$ and N is the number of states. In the discrete case, observation probabilities are stored in a $N \times M$ matrix, M being the number of symbols in the alphabet. In the continuous case, observation probabilities are modeled by mixture of multivariate Gaussians.

$$b_i(O_t) = \sum_{k=1}^{K} w_{ik} \mathcal{N}(O_t; \mu_{ik}, \Sigma_{ik}) \tag{A.1}$$

$$\mathcal{N}(O_t; \mu_{ik}, \Sigma_{ik}) = \frac{1}{\sqrt{(2\pi)^V |\Sigma_{ik}|}} e^{[\frac{-1}{2}(O_t - \mu_{ik})^2 \Sigma_{ik}^{-1} (O_t - \mu_{ik})^2]} \tag{A.2}$$

where μ_{ik}, Σ_{ik} are component means and covariances, respectively, and K denotes the number of components in the mixture. V is the dimensionality of the observation vectors and O_t is the observation at time t.

For a sequence classification problem, one is interested in evaluating the probability of any given observation sequence, $O_1 O_2 ... O_T$, given a HMM model, Θ. This probability, or the likelihood, $P(O|\Theta)$, of an HMM can be calculated in terms of the forward variable.

$$P(O|\Theta) = \sum_{i=1}^{N} \alpha_T(i) \tag{A.3}$$

where the forward variable, $\alpha_T(i)$, is the probability of observing the partial sequence $O_1 ... O_T$ until the end of the sequence, T, and being in state i at time T, given the

model Θ. The forward variable can be recursively calculated by going forward in time:

$$\alpha_1(j) = \pi_j b_j(O_1) \tag{A.4}$$

$$\alpha_t(j) = b_j(O_t) \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \tag{A.5}$$

For long sequences, the computation of the forward variable will exceed the precision range of the machine. Thus, a scaling procedure is needed to prevent underflow. The scaling coefficient, c_t is calculated as follows:

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)} \tag{A.6}$$

$$\hat{\alpha_t}(i) = c_t \alpha_t(i) \tag{A.7}$$

The computation of $P(O|\Theta)$ must be handled differently since $\alpha_T(i)$s are already scaled. $P(O|\Theta)$ can be calculated via the scaling coefficients. However we can only calculate the logarithm of P since P itself will be out of the precision range [121]:

$$log(P(O|\Theta)) = -\sum_{t=1}^{T} log c_t \tag{A.8}$$

The likelihood of an HMM can also be calculated in terms of both the forward and backward variables.

$$P(O|\Theta) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i) \tag{A.9}$$

where the forward variable, $\alpha_t(i)$, is 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 Θ and the backward variable, $\beta_t(i)$ is the probability of observing the partial sequence $O_{t+1}...O_T$ given that we are in state i at time t and the model Θ.

The backward variable can be recursively computed by going backwards:

$$\beta_T(i) = 1 \tag{A.10}$$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(O_{t+1}) \tag{A.11}$$

The probability of being in state i at time t given the model, Θ, can be computed by

$$\gamma_i(t) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} = \frac{\alpha_t(i)\beta_t(i)}{P(O|\Theta)} \tag{A.12}$$

and in the continuous case, the probability of the component k in state i at time t given the model can be computed by

$$\gamma_{ik}(t) = \frac{\alpha_t(i)\beta_t(i) w_{ik} N(O_t; \mu_{ik}, \Sigma_{ik})}{b_i(O_t) P(O|\Theta)} \tag{A.13}$$

and

$$\gamma_i(t) = \sum_{k=1}^{K} \gamma_{ik}(t) \tag{A.14}$$

A.2. Derivation of HMM Gradients

The gradient of the loglikelihood is given by

$$\frac{\partial ln P(O|\Theta)}{\partial \theta} = \frac{-1}{P(O|\Theta)} \frac{\partial P(O|\Theta)}{\partial \theta} \tag{A.15}$$

where θ stands for a parameter of the HMM: transition probabilities, a_{ij}, observation probabilities, $b_i(O_t)$, and in the case of continuous HMM, parameters of the component probabilities, $w_{ik}, \mu_{ik}, \Sigma_{ik}$.

A.2.1. Gradient with respect to Transition Probabilities

Using the chain rule in Equation A.9,

$$\begin{aligned}
\frac{\partial P(O|\Theta)}{\partial a_{ij}} &= \sum_{t=1}^{T} \frac{\partial P(O|\Theta)}{\partial \alpha_t i} \frac{\partial \alpha_t(i)}{\partial a_{ij}} + \sum_{t=1}^{T} \frac{\partial P(O|\Theta)}{\partial \beta_t i} \frac{\partial \beta_t(i)}{\partial a_{ij}} \\
&= \sum_{t=2}^{T} \beta_t(i) b_i(O_t) \alpha_{t-1}(j) + \sum_{t=1}^{T-1} \alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) \\
&= \sum_{t=1}^{T-1} [\beta_{t+1}(i) b_i(O_{t+1}) \alpha_t(j) + \beta_{t+1}(j) b_j(O_{t+1}) \alpha_t(i)] \qquad \text{(A.16)}
\end{aligned}$$

For left-to-right HMMs with no skips, using only self transition parameters will be enough. Hence, for $i = j$,

$$\begin{aligned}
\frac{\partial P(O|\Theta)}{\partial a_{ii}} &= \sum_{t=1}^{T-1} [\beta_{t+1}(i) b_i(O_{t+1}) \alpha_t(i) + \beta_{t+1}(i) b_i(O_{t+1}) \alpha_t(i)] \\
&= 2 \sum_{t=1}^{T-1} \beta_{t+1}(i) b_i(O_{t+1}) \alpha_t(i) \\
&\approx \sum_{t=1}^{T-1} \beta_{t+1}(i) b_i(O_{t+1}) \alpha_t(i) \qquad \text{(A.17)}
\end{aligned}$$

A.2.2. Gradient with respect to Observation Probabilities

Using the chain rule in Equation A.9,

$$\begin{aligned}
\frac{\partial P(O|\Theta)}{\partial b_i(O_t)} &= \frac{\partial P(O|\Theta)}{\partial \alpha_t i} \frac{\partial \alpha_t(i)}{\partial b_i(O_t)} + \frac{\partial P(O|\Theta)}{\partial \beta_t i} \frac{\partial \beta_t(i)}{\partial b_i(O_t)} \\
&= \beta_t(i) \frac{\alpha_t(i)}{b_i(O_t)} + \alpha_t(i) \frac{\partial \beta_t(i)}{\partial b_i(O_t)}
\end{aligned}$$

Since $\frac{\partial \beta_t(i)}{\partial b_i(O_t)} = 0$

$$\frac{\partial P(O|\Theta)}{\partial b_i(O_t)} = \beta_t(i) \frac{\alpha_t(i)}{b_i(O_t)} \qquad \text{(A.18)}$$

A.2.3. Gradient with respect to Component Probabilities

Using the chain rule in Equation A.9 and $\frac{\partial \beta_t(i)}{\partial b_i(O_t)} = 0$

$$\frac{\partial P(O|\Theta)}{\partial w_{ik}} = \sum_{t=1}^{T} \frac{\partial P(O|\Theta)}{\partial \alpha_t i} \frac{\partial \alpha_t(i)}{\partial b_i(O_t)} \frac{\partial b_i(O_t)}{\partial w_{ik}} \tag{A.19}$$

$$\frac{\partial P(O|\Theta)}{\partial \mu_{ik}} = \sum_{t=1}^{T} \frac{\partial P(O|\Theta)}{\partial \alpha_t i} \frac{\partial \alpha_t(i)}{\partial b_i(O_t)} \frac{\partial b_i(O_t)}{\partial \mu_{ik}} \tag{A.20}$$

$$\frac{\partial P(O|\Theta)}{\partial \Sigma_{ik}} = \sum_{t=1}^{T} \frac{\partial P(O|\Theta)}{\partial \alpha_t i} \frac{\partial \alpha_t(i)}{\partial b_i(O_t)} \frac{\partial b_i(O_t)}{\partial \Sigma_{ik}} \tag{A.21}$$

Then the partial derivatives can be calculated as

$$\frac{\partial b_i(O_t)}{\partial w_{ik}} = N(O_t; \mu_{ik}, \Sigma_{ik}) \tag{A.22}$$

$$\frac{\partial b_i(O_t)}{\partial \mu_{ik}} = N(O_t; \mu_{ik}, \Sigma_{ik})(O_t - \mu_{ik})^T \Sigma_{ik}^{-1} \tag{A.23}$$

$$\frac{\partial b_i(O_t)}{\partial \Sigma_{ik}} = N(O_t; \mu_{ik}, \Sigma_{ik})[-\Sigma_{ik}^{-1} + -\Sigma_{ik}^{-T}(O_t - \mu_{ik})(O_t - \mu_{ik})^T \Sigma_{ik}^{-T}] \tag{A.24}$$

Using equations A.15 and A.18, gradients of the loglikelihood can be calculated:

$$\frac{\partial lnp(O|\theta)}{\partial w_{ik}} = \sum_{t=1}^{T} \frac{\beta_t(i)\alpha_t(i)w_{ik}N(O_t; \mu_{ik}, \Sigma_{ik})}{P(O|\Theta)b_i(O_t)} \frac{1}{w_{ik}}$$

$$\frac{\partial lnp(O|\theta)}{\partial \mu_{ik}} = \sum_{t=1}^{T} \frac{\beta_t(i)\alpha_t(i)w_{ik}N(O_t; \mu_{ik}, \Sigma_{ik})}{P(O|\Theta)b_i(O_t)} (O_t - \mu_{ik})^T \Sigma_{ik}^{-1}$$

$$\frac{\partial lnp(O|\theta)}{\partial \Sigma_{ik}} = \sum_{t=1}^{T} \frac{\beta_t(i)\alpha_t(i)w_{ik}N(O_t; \mu_{ik}, \Sigma_{ik})}{P(O|\Theta)b_i(O_t)} [-\Sigma_{ik}^{-1} + -\Sigma_{ik}^{-T}(O_t - \mu_{ik})(O_t - \mu_{ik})^T \Sigma_{ik}^{-T}]$$

Using equation A.13,

$$\frac{\partial lnp(O|\theta)}{\partial w_{ik}} = \sum_{t=1}^{T} \frac{\gamma_{ik}(t)}{w_{ik}} \tag{A.25}$$

$$\frac{\partial lnp(O|\theta)}{\partial \mu_{ik}} = \sum_{t=1}^{T} \gamma_{ik}(t)(O_t - \mu_{ik})^T \Sigma_{ik}^{-1} \tag{A.26}$$

$$\frac{\partial lnp(O|\theta)}{\partial \Sigma_{ik}} = \sum_{t=1}^{T} \gamma_{ik}(t)[-\Sigma_{ik}^{-1} + -\Sigma_{ik}^{-T}(O_t - \mu_{ik})(O_t - \mu_{ik})^T \Sigma_{ik}^{-T}] \tag{A.27}$$

APPENDIX B: Belief Functions and the Transferable Belief Model

B.1. Belief Functions

In this section, we briefly present the necessary background on belief functions. Interested readers may refer to [95, 96, 97, 98] for more information on belief theories.

Frame: Let Ω be the set of N exclusive hypotheses: $\Omega = \{\Omega_1, \Omega_2, \ldots \Omega_N\}$. Ω is called the frame. It is the evidential counterpart of the probabilistic universe.

Powerset: Let 2^Ω, called the powerset of Ω, be the set of all the subsets A of Ω, including the empty set: $2^\Omega = \{A/A \subseteq \Omega\}$

Belief function (BF): A belief function is a set of scores defined on 2^Ω and adds up to 1, in exactly the same manner as a probability function (PF) defined on Ω. Let $m(.)$ be such a belief function. It represents our belief in the propositions that correspond to the elements of 2^Ω:

$$\begin{aligned} m : 2^\Omega &\rightarrow [0,1] \\ A &\mapsto m(A) \text{ with } \sum_{A \subseteq \Omega} m(A) = 1 \end{aligned}$$

A focal element is an element of the powerset to which a non-zero belief is assigned. Note that belief can be assigned to non-singleton propositions, which allows modeling the hesitation due to the absence of knowledge between elements.

Dempster's rule of combination: To combine several belief functions into a global belief function, one uses the Dempster's rule of combination. For N BFs, $m_1 \ldots m_N$, defined on the same hypothesis set Ω, the Dempster's rule of combination

is defined as:

$$\begin{aligned} \textcircled{\cap} : \mathfrak{B}^\Omega \times \mathfrak{B}^\Omega \times \ldots \times \mathfrak{B}^\Omega &\rightarrow \mathfrak{B}^\Omega \\ m_1 \textcircled{\cap} m_2 \textcircled{\cap} \ldots \textcircled{\cap} m_N &\mapsto m_{\textcircled{\cap}} \end{aligned}$$

with $\mathfrak{B}^\Omega$, the set of BFs defined on Ω, and $m_{\textcircled{\cap}}$, the global combined BF, which is calculated as:

$$m_{\textcircled{\cap}}(A) = \sum_{A=A_1 \cap \ldots \cap A_N} \left(\prod_{n=1}^{N} m_n(A_n) \right) \quad \forall A \in 2^\Omega \tag{B.1}$$

Decision making: After fusing several BFs, the knowledge is modeled via a single function over 2^Ω. There are alternative ways to make a decision from the knowledge of a BF [122, 123, 124]. A very popular method is to use the *Pignistic Transform* (PT), such as defined in the Transfer Belief Model [97], where the belief in doubtful focal elements is equally shared between the singleton hypotheses which are implied by them. Moreover, as the decision is supposed to be made within the defined hypotheses, the whole belief is normalized so that the belief in the empty set is not considered. One defines BetP(.) [97], the result of the PT of a BF $m(.)$, as:

$$BetP(h) = \frac{1}{1 - m(\emptyset)} \sum_{h \in A,\ A \subset \Omega} \frac{m(A)}{|A|} \quad \forall h \in \Omega \tag{B.2}$$

where $|A|$ denotes the cardinality of A. The division by $1 - m(\emptyset)$ is a normalizing factor. We will use the following notation of conditioning to express this normalization: $.|\Omega$.

In BF formalism, it is possible to make decisions based on other assumptions. For instance, it is possible to associate a plausibility to any element of the powerset, and then select the most plausible element [125, 126]. The plausibility of an element is the sum of all the belief associated with the hypothesis which fails to refute it. Hence, the bigger the cardinality of an element of the powerset is, the higher its plausibility is. Then, deciding according to the plausibility measure is likely to lead to a decision

on a set of hypotheses of high cardinality, including the entire Ω, which is finally an absence of decision. It may be wiser to prevent any decision making than making an inaccurate decision, especially in case of a decision with a huge cost of erring (juridic decision, for instance).

Basically, these two stances, to make a bet, or to wait for a cautious decision are typically opposed in decision making. For sign language recognition, we have the necessity to make a decision on which a reasonable mistake is acceptable, but one needs to reject a bet when excessive amount of information is missing: when classifying a sign with both manual and non-manual information, we accept a part of indecision on the non-manual gesture, but we have to make a complete decision on its manual part (see next sections). Hence, we need to use an intermediate way to make a decision: We need to precisely set the balance between cautiousness and the risk of a bet.

We propose to define a new method based on the PT. We generalize the PT so that we can decide whether any focal element has a too large cardinality with respect to the amount of uncertainty we allow, or, on the contrary, it is small enough (even if it is not a singleton focal element), to be considered. We call this transformation as Partial Pignistic Transform (PPT).

Partial Pignistic Transform: Let γ be an uncertainty threshold, and let $2^{|\Omega|_\gamma}$ be the set of all elements of the frame with cardinality smaller or equal to γ. We call $2^{|\Omega|_\gamma}$ the γ^{th} frame of decision.

$$2^{|\Omega|_\gamma} = \{A \in 2^\Omega / |A| \in \lfloor 0, \ldots, \gamma \rfloor\} \tag{B.3}$$

Let $M_\gamma(.)$ be the result by γ-PPT of a BF $m(.)$. It is defined on $2^{|\Omega|_\gamma}$ as:

$$
\begin{aligned}
M_\gamma(\emptyset) &= 0 \\
M_\gamma(A) &= m(A) + \sum_{B \supseteq A,\ B \notin 2^{|\Omega|_\gamma}} m(B) \frac{|A|}{\sum_{k=1}^{\gamma} \left[\binom{|B|}{k} \cdot k \right]}, \quad \forall A \in 2^{|\Omega|_\gamma} \backslash \emptyset
\end{aligned} \tag{B.4}
$$

where B are supersets of A, and $|A|$ denotes the cardinality of A. Then, the decision is made by simply choosing the most believable element of the γ^{th} frame of decision: $D^* = argmax_{2^{|\Omega|_\gamma}}(M_\gamma)$. Note that, the 1-PPT is equivalent to the PT.

B.2. Belief Function Definition from HMM Log-likelihoods

We define an Elemantary Belief Function (EBF) over the powerset of the two HMMs involved. Then, the belief of each EBF is distributed over one HMM, the other HMM, and the hesitation among the two HMMs. We modify this partition so that the HMM which has the smaller value among the two has a zero-valued belief. So, we simply define the repartition between one HMM and the union of the two involved. This is what we call the hesitation distribution, b, and it is modeled on the behavior of an expert human being. Then, for each pair of HMMs (HMM_i, HMM_j) an EBF m_{ij} is defined as

$$
\begin{aligned}
m_{ij}(\{i,j\}) &= b,\ b \in [0,1] \\
m_{ij}(\{i\}) &= 1 - b \\
m_{ij}(\{j\}) &= 0
\end{aligned} \tag{B.5}
$$

assuming that HMM_i gives a higher score than HMM_j. Then, the only point is to define the hesitation distribution, b, for each EBF.

We should keep in mind that, although the HMM scores are derived from probabilities, they are indeed log-likelihoods and it is not possible to compute the inverse and to go back to likelihoods because of the scaling operation carried out to prevent underflow of the probabilities when the sequences are long. Hence, if scaling is used,

only the log-likelihood is defined, and not the likelihood itself, due to the limits of machine representation of numbers [121].

This "conversion" problem is far more complicated when converting a real probability function into a belief function. The simple solution would be to convert the scores to probabilities by scaling and normalization operations and to remain in a "normalized" problem. However, this simple solution does not guarantee efficiency. Instead, we propose to fit a distribution to each pair of log-likelihoods and later combine them to produce belief values. We set this hesitation distribution experimentally and eventually tune it on a validation set. We use a simple model which assumes the belief in the hesitation distribution to follow a zero-mean Gaussian function with standard deviation, σ, with respect to the margins (the differences of scores). We define σ as $\sigma = \sqrt{\alpha \cdot \sigma_s}$, where σ_s is the variance of the margins of pairwise log-likelihoods for the HMM case. The coefficient α controls the level of uncertainty in the belief function. The bigger it is, the more hesitation the belief function contains. If α is too small, the belief function will be equivalent to a max function over the likelihoods: $argmax(\mathcal{L}(.))$ will focus all the belief, and the rest will be zero-believed ($\mathcal{L}(.)$ denotes the likelihood of each model). On the contrary, if α is too big, the belief is focused on the widest hypothesis (the complete hesitation), and making a non-random decision over such a function is impossible.

Having defined the hesitation distribution, the EBFs are calculated and they are fused together with Dempster's rule of combination:

$$m = \bigcirc\!\!\!\!\!\!\cap_{i,j\neq i}\, m_{ij} \tag{B.6}$$

The entire algorithm to compute a belief function from a set of nonhomogeneous scores is given in Figure B.1. So far, we described how to obtain the hesitation distribution for each EBF, and the influence of these models on the global belief function (step 4 of Figure B.1), but we have not defined how to create this global belief function from the EBFs. This is done by (1) refining the EBFs so that they are defined on the

complete powerset of Ω instead of on a part of it [6](see step 4 of Figure B.1), and (2) fusing all the EBFs with Dempster's rule of combination (step 5 of Figure B.1).

Let Ω be the set of all the hypotheses (or classes): $\Omega = \{H_1, H_2, \ldots H_N\}$

1. For all pairs, (H_i, H_j), in Ω, where $i \neq j$, compute $V_{ij} = \mathcal{L}(H_i) - \mathcal{L}(H_j)$.
2. Compute σ_s of V_{ij} for all (H_i, H_j) pairs.
3. Set α to define the level of hesitation of the hesitation distribution.
4. For each V_{ij}, create a belief function $m_{ij}(.)$ over the powerset of $\{H_i, H_j\}$. This BF defines the belief on $\{H_i\}$, $\{H_j\}$ and $\{H_i, H_j\}$, with zero-belief for $\emptyset$ and one of $\{H_i\}$ or $\{H_j\}$. Then refine the EBFs to make them combinable :

$$m_{ij}(\{i,j\}) = \exp\left(\frac{-V_{ij}^2}{2*(\alpha*\sigma_s)^2}\right) \quad , \quad m_{ij}(\{i,j\}) \xrightarrow{\textit{refinement}} m_{ij}(\{\Omega\})$$

$$m_{ij}(\{i\}) = \begin{cases} 1 - m(\{i,j\}) & \text{if } V_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \quad , \quad m_{ij}(\{i\}) \xrightarrow{\textit{refinement}} m_{ij}(\{\Omega \backslash j\})$$

$$m_{ij}(\{j\}) = \begin{cases} 1 - m(\{i,j\}) & \text{if } V_{ij} < 0 \\ 0 & \text{otherwise} \end{cases} \quad , \quad m_{ij}(\{j\}) \xrightarrow{\textit{refinement}} m_{ij}(\{\Omega \backslash i\})$$

5. Apply Dempster's rule of combination to these belief functions $m_{ij}(.)$.
6. The result is the combined belief function $m_{\bigcirc}$ over Ω on which a decision can be made (Equation B.1).

Figure B.1. Algorithm to compute beliefs from a set of nonhomogeneous scores.

REFERENCES

1. Aran, O. and L. Akarun, "A Particle Filter Based Algorithm for Robust Tracking of Hands and Face Under Occlusion", *IEEE 16th Signal Processing and Communications Applications (SIU 2008)*, 2008.

2. Aran, O., I. Ari, A. Benoit, P. Campr, A. H. Carrillo, F.-X. Fanard, L. Akarun, A. Caplier, M. Rombaut, and B. Sankur, "SignTutor: An Interactive System for Sign Language Tutoring", *IEEE Multimedia*, 2008, , to appear.

3. Aran, O., T. Burger, A. Caplier, and L. Akarun, "Sequential Belief Based Fusion of Manual and Non-manual Signs", *Gesture Workshop*, May 2007.

4. Aran, O., T. Burger, A. Caplier, and L. Akarun, "A Belief-Based Sequential Fusion Approach for Fusing Manual and Non-Manual Signs", *Pattern Recognition*, 2008 (in press).

5. Aran, O., T. Burger, L. Akarun, and A. Caplier, *Multimodal user interfaces: from signals to interaction*, chapter Gestural Interfaces for Hearing-Impaired Communication, pp. 219–250, Springer, 2008.

6. Burger, T., O. Aran, and A. Caplier, "Modeling Hesitation and Conflict: A Belief-Based Approach for Multi-class Problems", *ICMLA '06: Proceedings of the 5th International Conference on Machine Learning and Applications*, pp. 95–100, IEEE Computer Society, Washington, DC, USA, 2006.

7. Burger, T., A. Urankar, O. Aran, L. Akarun, and A. Caplier, "Cued Speech Hand Shape Recognition", *2nd International Conference on Computer Vision Theory and Applications (VISAPP'07), Spain*, 2007.

8. Aran, O. and L. Akarun, "Recognizing Two Handed Gestures with Generative, Discriminative and Ensemble Methods via Fisher Kernels", *LNCS: Multime-*

dia Content Representation, Classification and Security International Workshop (MRCS), Vol. 4015, pp. 159–166, 2006.

9. Aran, O. and L. Akarun, "Multi-class Classification Strategies for Fisher Scores of Gesture and Sign Sequences", *International Conference On Pattern Recognition*, 2008.

10. 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*, 2008.

11. Kendon, A., "Current issues in the study of gesture", Nespoulous, J., P. Peron, and A. R. Lecours (editors), *The Biological Foundations of Gestures: Motor and Semiotic Aspects*, pp. 23–47, Lawrence Erlbaum Assoc, 1986.

12. McNeill, D. and E. Levy, "Conceptual Representations in Language Activity and Gesture", Jarvella and Klein (editors), *Speech, Place, and Action*, John Wiley and Sons Ltd, 1982.

13. Wu, Y. and T. S. Huang, "Vision-Based Gesture Recognition: A Review", *Lecture Notes in Computer Science*, Vol. 1739, No. 1, pp. 103+, 1999.

14. Quek, F. K. H., "Eyes in the interface.", *Image and Vision Computing*, Vol. 13, No. 6, pp. 511–525, 1995.

15. Quek, F. K. H., "Toward a Vision-Based Hand Gesture Interface", Singh, G., S. K. Feiner, and D. Thalmann (editors), *Virtual Reality Software and Technology: Proceedings of the VRST'94 Conference*, pp. 17–31, World Scientific, London, 1994.

16. Wu, Y. and T. S. Huang, "Hand Modeling, Analysis, and Recognition for Vision Based Human Computer Interaction", *IEEE Signal Processing Magazine*, Vol. 21, No. 1, pp. 51–60, 2001.

17. Pavlovic, V., R. Sharma, and T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 677–695, 1997.

18. McNeil, D., *Hand and Mind: What Gestures Reveal about Thought*, The University of Chicago Press, 1992.

19. Kendon, A., *Gesture*, Cambridge, 2004.

20. Miles, M., "Signing in the Seraglio: mutes, dwarfs and jesters at the Ottoman Court 1500–1700", *Disability & Society*, Vol. 15, No. 1, pp. 115–134, 2000.

21. Stokoe, W. C., "Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf", *Studies in Linguistics: Occasional papers*, Vol. 8, 1960.

22. Liddell, S. K., *Grammar, Gesture, and Meaning in American Sign Language*, Cambridge University Press, 2003.

23. Cornett, R. O., "Cued Speech", *American Annals of the deaf*, Vol. 112, pp. 3–13, 1967.

24. Foulds, R., "Biomechanical and perceptual constraints on the bandwidth requirements of sign language", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 12, No. 1, pp. 65–72, 2004.

25. Manoranjan, M. and J. Robinson, "Practical low-cost visual communication using binary images for deaf sign language", *IEEE Transactions on Rehabilitation Engineering*, Vol. 8, No. 1, pp. 81–88, 2000.

26. Sperling, G., "Video Transmission of American Sign Language and Finger Spelling: Present and Projected Bandwidth Requirements", *IEEE Transactions on Communications*, Vol. 29, No. 12, pp. 1993– 2002, 1981.

27. Chiu, Y.-H., H.-Y. Su, and C.-J. Cheng, "Joint Optimization of Word Alignment and Epenthesis Generation for Chinese to Taiwanese Sign Synthesis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 1, pp. 28–39, 2007.

28. Karpouzis, K., G. Caridakis, S. E. Fotinea, and E. Efthimiou, "Educational resources and implementation of a Greek sign language synthesis architecture", *Computers and Education*, Vol. 49, No. 1, pp. 54–74, 2007.

29. Ohene-Djan, J. and S. Naqvi, "An Adaptive WWW-Based System to Teach British Sign Language", *ICALT '05: Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies*, pp. 127–129, IEEE Computer Society, Washington, DC, USA, 2005.

30. Wu, C.-H., Y.-H. Chiu, and K.-W. Cheng, "Error-Tolerant Sign Retrieval Using Visual Features and Maximum A Posteriori Estimation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 4, pp. 495–508, 2004.

31. Tyrone, M. E., "Overview of Capture Techniques for Studying Sign Language Phonetics.", *Gesture Workshop*, pp. 101–104, 2001.

32. Awad, G., J. Han, and A. Sutherland, "A Unified System for Segmentation and Tracking of Face and Hands in Sign Language Recognition", *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pp. 239–242, IEEE Computer Society, Washington, DC, USA, 2006.

33. Holden, E.-J., G. Lee, and R. Owens, "Australian sign language recognition", *Machine Vision and Applications*, Vol. 16, No. 5, pp. 312–320, 2005.

34. Habili, N., C.-C. Lim, and A. Moini, "Segmentation of the face and hands in sign language video sequences using color and motion cues.", *IEEE Trans. Circuits Syst. Video Techn.*, Vol. 14, No. 8, pp. 1086–1097, 2004.

35. Imagawa, I., H. Matsuo, R. Taniguchi, D. Arita, S. Lu, and S. Igi, "Recognition of Local Features for Camera-Based Sign Language Recognition System", *ICPR '00: Proceedings of the International Conference on Pattern Recognition*, Vol. 4, pp. 849–853, IEEE Computer Society, Washington, DC, USA, 2000.

36. Cui, Y. and J. Weng, "A Learning-Based Prediction-and-Verification Segmentation Scheme for Hand Sign Image Sequence.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 8, pp. 798–804, 1999.

37. Hamada, Y., N. Shimada, and Y. Shirai, "Hand Shape Estimation under Complex Backgrounds for Sign Language Recognition", *Proc. of 6th Int. Conf. on Automatic Face and Gesture Recognition*, pp. 589 – 594, 2004.

38. Ong, E.-J. and R. Bowden, "A Boosted Classifier Tree for Hand Shape Detection.", *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 889–894, 2004.

39. Kalman, R. E., "A new approach to linear filtering and prediction problems", *Transactions of the ASME–Journal of Basic Engineering*, Vol. 82, pp. 35–45, 1960.

40. Imagawa, K., S. Lu, and S. Igi, "Color-Based Hands Tracking System for Sign Language Recognition", *3rd International Conference on Face and Gesture Recognition*, p. 462, 1998.

41. Isard, M. and A. Blake, "Condensation – conditional density propagation for visual tracking", *International Journal of Computer Vision*, Vol. 26, No. 1, pp. 5–28, 1998.

42. Dreuw, P., T. Deselaers, D. Rybach, D. Keysers, and H. Ney, "Tracking Using Dynamic Programming for Appearance-Based Sign Language Recognition", *7th International Conference on Face and Gesture Recognition*, pp. 293–298, 2006.

43. Hienz, H. and K. Grobel, "Automatic estimation of body regions from video images", *Gesture Workshop*, 1997.

44. Cooper, H. and R. Bowden, "Large Lexicon Detection of Sign Language", *HCI*, pp. 88–97, 2007.

45. Yang, X., F. Jiang, H. Liu, H. Yao, W. Gao, and C. Wang, "Visual Sign Language Recognition Based on HMMs and Auto-regressive HMMs", *Gesture Workshop*, Vol. 3881 of *Lecture Notes in Computer Science*, pp. 80–83, 2005.

46. Bowden, R., D. Windridge, T. Kadir, A. Zisserman, and M. Brady, "A Linguistic Feature Vector for the Visual Interpretation of Sign Language", *8th European Conference on Computer Vision, Prague, Czech Republic*, Springer, May 2004.

47. Chang, C.-C. and C.-M. Pengwu, "Gesture recognition approach for sign language using curvature scale space and hidden Markov model", *ICME '04: IEEE International Conference on Multimedia and Expo*, Vol. 2, pp. 1187– 1190, 2004.

48. Yang, M. H., N. Ahuja, and M. Tabb, "Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 8, pp. 1061–1074, 2002.

49. Kadir, T., R. Bowden, E. Ong, and A. Zisserman, "Minimal Training, Large Lexicon, Unconstrained Sign Language Recognition", *15th British Machine Vision Conference, Kingston*, 2004.

50. Munib, Q., M. Habeeba, B. Takruria, and H. A. Al-Malika, "American sign language (ASL) recognition based on Hough transform and neural networks", *Expert Systems with Applications*, Vol. 32, No. 1, pp. 24–37, January 2007.

51. Al-Jarrah, O. and A. Halawani, "Recognition of gestures in Arabic sign language using neuro-fuzzy systems", *Artificial Intelligence*, Vol. 133, No. 1-2, pp. 117–138, 2001.

52. Wu, J. and W. Gao, "The Recognition of Finger-Spelling for Chinese Sign Language", *Gesture Workshop*, pp. 96–100, 2001.

53. Ong, S. C., S. Ranganath, and Y. Venkatesha, "Understanding gestures with systematic variations in movement dynamics", *Pattern Recognition*, Vol. 39, No. 9, pp. 1633–1648, 2006.

54. Sagawa, H., M. Takeuchi, and M. Ohki, "Methods to describe and recognize sign language based on gesture components represented by symbols and numerical values", *Knowledge-Based Systems, Intelligent User Interfaces*, Vol. 10, No. 5, pp. 287–294, 1998.

55. Hu, M. K., "Visual pattern recognition by moment invariants", *IRE Transactions on Information Theory*, Vol. 8, pp. 179–187, 1962.

56. Caplier, A., L. Bonnaud, S. Malassiotis, and M.G.Strintzis, "Comparison of 2D and 3D analysis for automated Cued Speech gesture recognition", *SPECOM*, 2004.

57. Shimada, N., K. Kimura, and Y. Shirai, "Real-time 3-D Hand Posture Estimation based on 2-D Appearance Retrieval Using Monocular Camera", *Proc. Int. WS. on RATFG-RTS (satellite WS of ICCV2001)*, pp. 23 – 30, 2001.

58. Ong, S. C. W. 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, 2005.

59. Zeshan, U., "Aspects of Trk Isaret Dili (Turkish Sign Language)", *Sign Language & Linguistics*, Vol. 6, No. 1, pp. 43–75, 2003.

60. Ma, J., W. Gao, and R. Wang, "A Parallel Multistream Model for Integration of Sign Language Recognition and Lip Motion", *Third International Conference on Advances in Multimodal Interfaces (ICMI '00)*, pp. 572–581, Springer-Verlag,

London, UK, 2000.

61. Erdem, U. and S. Sclaroff, "Automatic Detection of Relevant Head Gestures in American Sign Language Communication", *International Conference on Pattern Recognition*, Vol. 1, pp. 460–463, 2002.

62. Xu, M., B. Raytchev, K. Sakaue, O. Hasegawa, A. Koizumi, M. Takeuchi, and H. Sagawa, "A Vision-Based Method for Recognizing Non-manual Information in Japanese Sign Language", *Third International Conference on Advances in Multimodal Interfaces (ICMI '00)*, pp. 572–581, Springer-Verlag, London, UK, 2000.

63. Ming, K. and S. Ranganath, "Representations for Facial Expressions", *International Conference on Control Automation, Robotics and Vision*, Vol. 2, pp. 716–721, 2002.

64. Starner, T. and A. Pentland, "Real-Time American Sign Language Recognition From Video Using Hidden Markov Models", *SCV95*, 1995.

65. Vogler, C. and D. Metaxas, "Adapting Hidden Markov models for ASL recognition by using three-dimensional computer vision methods", *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 156–161, 1997.

66. Vogler, C. and D. Metaxas, "ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis", *Sixth International Conference on Computer Vision (ICCV '98)*, p. 363, IEEE Computer Society, Washington, DC, USA, 1998.

67. Fang, G., W. Gao, X. Chen, C. Wang, and J. Ma, "Signer-Independent Continuous Sign Language Recognition Based on SRN/HMM", *Gesture Workshop*, pp. 76–85, 2001.

68. Vogler, C. and D. N. Metaxas, "Handshapes and Movements: Multiple-Channel American Sign Language Recognition.", *Gesture Workshop*, pp. 247–258, 2003.

69. Vogler, C. and D. Metaxas, "Parallel Hidden Markov Models for American Sign

Language Recognition", *International Conference on Computer Vision, Kerkyra, Greece*, Vol. 1, pp. 116–122, 1999.

70. Fang, G., W. Gao, and D. Zhao, "Large vocabulary sign language recognition based on fuzzy decision trees.", *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, Vol. 34, No. 3, pp. 305–314, 2004.

71. Kadous, M. W., "Machine Recognition of Auslan Signs Using Powergloves: Towards Large-lexicon Recognition of Sign Languages,", *Workshop on the Integration of Gestures in Language and Speech, Wilmington Delaware*, 1996.

72. Zhang, L.-G., X. Chen, C. Wang, Y. Chen, and W. Gao, "Recognition of sign language subwords based on boosted hidden Markov models.", *ICMI*, pp. 282–287, 2005.

73. Zahedi, M., D. Keysers, and H. Ney, "Pronunciation Clustering and Modeling of Variability for Appearance-Based Sign Language Recognition.", *Gesture Workshop*, pp. 68–79, 2005.

74. Sarfraz, M., Y. A. Syed, and M. Zeeshan, "A System for Sign Language Recognition Using Fuzzy Object Similarity Tracking", *IV '05: Proceedings of the Ninth International Conference on Information Visualisation (IV'05)*, pp. 233–238, IEEE Computer Society, Washington, DC, USA, 2005.

75. Assan, M. and K. Grobel, "Video-Based Sign Language Recognition Using Hidden Markov Models.", *Gesture Workshop*, pp. 97–109, 1997.

76. Murakami, K. and H. Taguchi, "Gesture recognition using recurrent neural networks", *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 237–242, ACM, New York, NY, USA, 1991.

77. Liddell, S. K. and R. E. Johnson, "American Sign Language: The phonological base", *Sign Language Studies*, Vol. 64, pp. 195–278, 1989.

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

79. Wang, C., X. Chen, and W. Gao, "A Comparison Between Etymon- and Word-Based Chinese Sign Language Recognition Systems.", *Gesture Workshop*, pp. 84–87, 2005.

80. Fang, G., X. Gao, W. Gao, and Y. Chen, "A Novel Approach to Automatically Extracting Basic Units from Chinese Sign Language", *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pp. 454–457, IEEE Computer Society, Washington, DC, USA, 2004.

81. Vogler, C. and D. N. Metaxas, "Toward Scalability in ASL Recognition: Breaking Down Signs into Phonemes", *Gesture Workshop*, pp. 211–224, Springer-Verlag, London, UK, 1999.

82. Marcel, S. and A. Just, "IDIAP Two Handed Gesture Dataset", Available at http://www.idiap.ch/∼marcel/.

83. Wilbur, R. B. and A. C. Kak, "Purdue RVL-SLLL American Sign Language Database", Techincal report TR-06-12, School of Electrical and Computer Engineering, Purdue University, W. Lafayette, IN 47906., 2006.

84. Athitsos, V., C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan, and A. Thangali, "The ASL lexicon video dataset", *IEEE Workshop on CVPR for Human Communicative Behavior Analysis*, 2008.

85. Edwards, A. D. N., "Progress in Sign Languages Recognition.", *Gesture Workshop*, pp. 13–21, 1997.

86. Reid, D., "An algorithm for tracking multiple targets", *IEEE Transactions on Automatic Control*, Vol. 24, No. 6, pp. 843–854, 1979.

87. Yeasin, M., E. Polat, and R. Sharma, "A Multiobject Tracking Framework for Interactive Multimedia Applications", *IEEE Transactions on MultiMedia*, Vol. 6, No. 3, pp. 398–405, 2004.

88. Doucet, A., N. De Freitas, and N. Gordon (editors), *Sequential Monte Carlo methods in practice*, Springer-Verlag, 2001.

89. Isard, M. and J. MacCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker", *International Conference on Computer Vision(ICCV'01)*, pp. II: 34–41, 2001.

90. Czyz, J., B. Ristic, and B. Macq, "A particle filter for joint detection and tracking of color objects", *Image and Vision Computing*, Vol. 25, No. 8, pp. 1271–1281, August 2007.

91. Kang, H. and D. Kim, "Real-time multiple people tracking using competitive condensation", *Pattern Recognition*, Vol. 38, No. 7, pp. 1045–1058, July 2005.

92. Khan, Z., T. Balch, and F. Dellaert, "MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 11, pp. 1805–1918, 2005.

93. Maggio, E. and A. Cavallaro, "Hybrid Particle Filter and Mean Shift tracker with adaptive transition model", *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.

94. Vezhnevets, V., V. Sazonov, and A. Andreeva, "A Survey on Pixel-based Skin Color Detection Techniques", *Graphicon*, pp. 85–92, 2003.

95. Dempster, A. P., "A Generalization of Bayesian Inference", *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 30, No. 2, pp. 205–247, 1968.

96. Shafer, G., *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

97. Smets, P. and R. Kennes, "The transferable belief model", *Artificial Intelligence*,

Vol. 66, No. 2, pp. 191–234, 1994.

98. Kohlas, J. and P. A. Monney, "Theory of Evidence: A Survey of its Mathematical Foundations, Applications and Computations", *ZOR-Mathematical Methods of Operational Research*, Vol. 39, pp. 35–68, 1994.

99. Benoit, A. and A. Caplier, "Head Nods analysis: Interpretation of non verbal communication gestures", *International Conference on Image Processing, ICIP2005, Genova, Italy*, Vol. 3, pp. 425–428, 2005.

100. Jaakkola, T. S. and D. Haussler, "Exploiting generative models in discriminative classifiers", *Conference on Advances in Neural Information Processing Systems II*, pp. 487–493, MIT Press, 1998.

101. Smith, N. and M. Gales, "Speech recognition using SVMs", Dietterich, T., S. Becker, and Z. Ghahramani (editors), *Advances in Neural Information Processing Systems.*, Vol. 14, MIT Press, 2002.

102. Jaakkola, T., M. Diekhans, and D. Haussler, "Using the Fisher Kernel Method to Detect Remote Protein Homologies", *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pp. 149–158, AAAI Press, 1999.

103. Moreno, P. and R. Rifkin, "Using the Fisher kernel method for Web audio classification", *IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP '00*, Vol. 6, pp. 2417–2420, 2000.

104. Holub, A. D., M. Welling, and P. Perona, "Combining Generative Models and Fisher Kernels for Object Recognition", *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, Vol. 1, pp. 136–143, IEEE Computer Society, Washington, DC, USA, 2005.

105. Chen, L. and H. Man, "Hybrid IMM/SVM approach for wavelet-domain proba-

bilistic model based texture classification", *IEE Proceedings of Vision, Image and Signal Processing*, Vol. 152, No. 6, pp. 724–730, 2005.

106. Chen, L., H. Man, and A. V. Nefian, "Face Recognition based on multi-class mapping of Fisher scores", *Pattern Recognition*, Vol. 38, pp. 799–811, 2005.

107. He, X., L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition", *Signal Processing Magazine, IEEE*, Vol. 25, No. 5, pp. 14–36, September 2008.

108. Cuturi, M., J.-P. Vert, O. Birkenes, and T. Matsui, "A Kernel for Time Series Based on Global Alignments", *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, Vol. 2, pp. II–413–II–416, April 2007.

109. Smith, N. and M. Gales, "Using SVMs to Classify Variable Length Speech Patterns", Technical report, Cambridge University Engineering Department, 2002.

110. Liu, N., B. C. Lovell, P. J. Kootsookos, and R. I. A. Davis, "Model Structure Selection and Training Algorithms for an HMM Gesture Recognition System", *Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR'04)*, pp. 100–105, IEEE Computer Society, Washington, DC, USA, 2004.

111. Duda, R. O., P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, Inc., 2001.

112. Pudil, P., J. Novovicova, and J. Kittler, "Floating search methods in feature selection", *Pattern Recognition Letters*, Vol. 15, No. 11, pp. 1119–1125, 1994.

113. Aran, O., I. Ari, P. Campr, E. Dikici, M. Hruz, D. Kahramaner, S. Parlak, L. Akarun, and M. Saraclar, "Speech and Sliding Text Aided Sign Retrieval from Hearing Impaired Sign News Videos", *eNTERFACE'07 The Summer Workshop on Multimodal Interfaces, Istanbul, Turkey*, 2007.

114. "SignTutor demonstration video", `http://www.cmpe.boun.edu.tr/pilab/pilabfiles/demos/signtutor_demo_DIVX.avi`.

115. Jayaram, S., S. Schmugge, M. C. Shin, and L. V. Tsap, "Effect of color space transformation, the illuminance component, and color modeling on skin detection", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*, pp. 813–818, 2004.

116. Beaudot, W., *The neural information processing in the vertebrate retina: A melting pot of ideas for artificial vision*, Ph.D. thesis, INPG (France), December 1994.

117. Torralba, A. and J. Herault, "An efficient neuromorphic analog network for motion estimation", *IEEE Transactions on Circuits and Systems-I: Special Issue on Bio-Inspired Processors and CNNs for Visio*, Vol. 46, No. 2, 1999.

118. Pandzic, I. and R. Forchheimer, *MPEG-4 facial animation: The standard, implementation and applications*, Wiley, 2002.

119. Balci, K., "Xface: Open source toolkit for creating 3D faces of an embodied conversational agent", *5th International Symposium SmartGraphics*, 2005.

120. Parlak, S. and M. Saraclar, "Spoken Term Detection for Turkish Broadcast News", *ICASSP*, 2008.

121. Rabiner, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, Vol. 77 of *no. 2*, pp. 257–285, February 1989.

122. Cobb, B. and P. Shenoy, "A Comparison of Methods for Transforming Belief Functions Models to Probability Models", *Lecture Notes in Artificial Intelligence*, Vol. 2711, pp. 255–266, 2003.

123. Daniel, M., "Probabilistic Transformations of Belief Functions", *ECSQARU*, pp. 539–551, 2005.

124. Smets, P., *Decision making in a context where uncertainty is represented by belief functions*, chapter Belief Functions in Business Decisions, pp. 17–61, Physica-Verlag, Heidelberg, Germany, IRIDIA, 2002.

125. Shafer, G. and P. P. Shenoy, "Local computation in hypertrees", Technical report, School of Business, University of Kansas, 1991.

126. Cobb, B. R. and P. P. Shenoy, "On the plausibility transformation method for translating belief function models to probability models.", *International Journal of Approximate Reasoning*, Vol. 41, No. 3, pp. 314–330, 2006.