A FACE TRACKING SYSTEM FOR DYNAMIC EVENT RECOGNITION: APPLICATION TO CONTINUOUS RECOGNITION OF NON-MANUAL MARKERS OF AMERICAN SIGN LANGUAGE AND TO DECEPTION DETECTION BY KINESIC ANALYSIS

BY NICHOLAS MICHAEL

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Professor Dimitris N. Metaxas

and approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

January, 2012

ABSTRACT OF THE DISSERTATION

A Face Tracking System for Dynamic Event Recognition: Application to Continuous Recognition of Non-Manual Markers of American Sign Language and to Deception Detection by Kinesic Analysis

by Nicholas Michael

Dissertation Director: Professor Dimitris N. Metaxas

Face tracking has numerous applications in the field of Human Computer Interaction and behavior understanding in general. Yet, face tracking is a difficult problem because the tracker must generalize to new faces, adapt to changing illumination, keep up with fast motions and pose changes, and tolerate target occlusion. We first present our efforts to develop a system for probabilistic face tracking, using anthropometric and appearance constraints. We then move onto the focus of our work, which is the application of the face tracker to two interesting recognition problems.

Firstly, given that sign language is used as a primary means of communication by deaf individuals and as augmentative communication by hearing individuals with a variety of disabilities, the development of robust, real-time sign language recognition technologies would be a major step forward in making computers equally accessible to everyone. However, most research in the field of sign language recognition has focused on the manual component of signs, despite the fact that there is critical grammatical information expressed through facial expressions and head gestures. Therefore, we present our novel framework for robust tracking and analysis of facial expressions and

head gestures, by means of a dynamic feature descriptor, a 3D face model and temporal models, with an application to sign language recognition. We apply it to successful continuous recognition of six different classes of non-manual grammatical expressions.

Secondly, deception is present in our everyday social and professional lives and its detection can be beneficial, not only to us individually but to our society as a whole. For example, accurate deception detection can aid law enforcement officers in solving a crime. It can also help border control agents to detect potentially dangerous individuals during routine screening interviews. Therefore, we also present two novel methods for deception detection, using only visual cues extracted from our face tracker and a skin blob tracker, both with promising results. One is based on a novel kernel density descriptor of human behavior, which can differentiate normal behavior profiles from over-controlled and agitated ones, using nearest neighbor search. The other is based on the notion of subject-interviewer synchrony.

Acknowledgements

I am grateful to Professor Dimitris Metaxas for his advice and support during my PhD studies. He has been a motivating and encouraging figure, as well as a never ending source of knowledge. None of the work in this thesis would have been possible without him.

I want to thank the other members of my examination committee: Prof. Vladimir Pavlovic, Prof. Tina Eliassi-Rad and Prof. Ioannis Stamos (CUNY) for their advice, help and valuable suggestions regarding this thesis. It is a privilege to have each of them serve in my examination committee.

I also need to thank all my other professors at Rutgers (in alphabetical order) who, through their courses, contributed to my success by enriching my brain with valuable knowledge and skills, which I applied directly or indirectly to my research: Prof. Doug Decarlo, Prof. Ahmed Elgammal, Prof. Michael Fredman, Prof. Kathleen M. Goelz, who supervised me while I was a Teaching Assistant, Prof. Rebecka Jörnsten, Prof. Bahman Kalantari, Prof. Michael Littman, Prof. Amelie Marian and Prof. Gerard Richter.

Special thanks to all other professors and researchers with whom I collaborated to produce the work presented in this thesis, as well as work presented or published elsewhere. More specifically, I thank Prof. Carol Neidle (Boston University) for her collaboration and assistance on the Sign Language project. I thank Prof. Judee K. Burgoon (University of Arizona), Prof. Norah Dunbar and Prof. Matthew Jensen (Oklahoma University) for their collaboration and assistance on the Credibility Assessment and Deception Detection projects. I thank Prof. David F. Dinges (University of Pennsylvania) for his collaboration and assistance on the Optical Computer Recognition projects for stress and emotion analysis in space flight. I thank Prof. Marsha Bates

(Rutgers University) for her collaboration and assistance on the project which explored the effects of alcohol on facial expressions and emotions.

I want to give many thanks to all my friends and colleagues at CBIM. It was a pleasure to engage with them in fruitful discussions about various research topics on a regular basis and to occasionally philosophize life. Their presence has really enriched my experience at Rutgers.

Lastly, I acknowledge that in the course of my research, while making progress towards my PhD, I have published to various conferences and workshops a large portion of the work, which I collectively present in a unified fashion in this thesis [16, 74, 75, 76, 77, 84].

Dedication

This dissertation is dedicated to my beloved wife, my other half, who has patiently waited for this moment from afar for six long years: Elena Michael.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The human face is a crucial channel of non-verbal communication. Through static facial signs [32], such as bone structure, eye shape and color, etc., which constitute a person's appearance, we are able to identify and distinguish people from each other. Through dynamic facial signs [32], such as their facial expressions, humans can convey information about their emotional state (e.g., happiness, sadness, surprise, fear), general nervousness, their interest or disinterest in something and so on [33, 34, 35, 36, 37]. If human computer interaction is to be successful, computers need to be able to understand non-verbal dynamic behaviors. Therefore, what is needed is a system which can tap into this important communication channel and through accurate face tracking (despite variations in facial shapes, illumination, head pose and facial expressions) and dynamic event modelling, recognize this plethora of information that people convey non-verbally.

In this thesis, we present our work on a face tracking system, which can track and recognize such dynamic facial signals, as well as other non-verbal dynamic behaviors. Our system is built on the work of Kanaujia et al. [57] who model the facial shape manifold by a piecewise linear approximation. We present a particle filter extension [51] to their face tracker with hierarchical observation likelihoods, which enforce implicit anthropometric and multi-level appearance constraints for more accurate tracking. The improvement in tracking accuracy is shown quantitatively on a very challenging video sequence with facial occlusion. We then proceed to describe the successful application of our system and methods to two novel recognition problems: (1) recognition of the non-manual component of American Sign Language (ASL), (2) detection of deception in interview scenarios using only visual cues through kinesic and synchrony analysis.

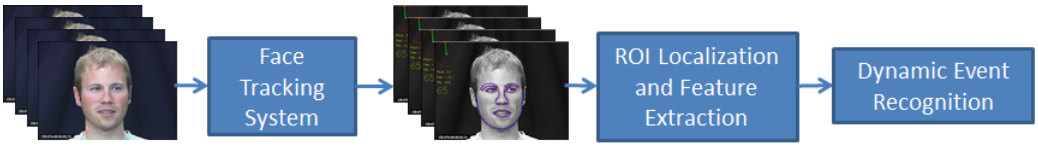

Figure 1.1: Illustration of our system's framework.

The overall framework we use in both applications is shown in Fig. 1.1. The input video is tracked by the face tracking system. Based on the tracked result, the system localizes a Region of Interest (ROI), which can be application-specific. For example, for recognizing non-manual markers in ASL we focus on the eyes, eyebrows and nose, while for deception detection we also look at the mouth region. It then extracts discriminative features characterizing the dynamic nature of the events we want to recognize and using a trained model the system outputs a prediction. In the following sections of this chapter we provide additional details motivating our face tracking system and each of its applications.

1.1.1 Face Tracking

Face tracking is an important component of any computerized human interaction system. It is also the building block of many applications in biometrics, facial expression analysis and synthesis. For these applications, it is not enough to track a bounding box around the face. Instead, the face tracker should also localize the different facial components, e.g., eyes, nose, mouth, eyebrows, and it must do so across multiple views. This is a challenging problem because although human faces share a common structure (i.e., the facial components mentioned above), there is still significant variability across different individuals. Yet, what is required it a face tracker, which once trained, can generalize well to unseen faces and handle illumination changes. It should also cope with partial occlusions and pose changes, such as head rotations, which cause drastic changes in the shape of the face, causing it to lie on a non-linear manifold [57, 80]. As the head rotates by a certain amount, the shape of the different parts of the face, as viewed from a two dimensional perspective, does not change uniformly and by an equal amount in all places. This effect is more severe during head rotations which approach

profile poses.

Some methods rely on parametrized 3D deformable models [22, 23, 24, 44, 62, 116]. The accuracy of these methods depends on the correct estimation of the image features, which are used to estimate the parameters of the 3D model. Some feature extraction methods use deterministic tracking [100], while others track feature distributions (e.g., [51]). The accuracy of these methods can be affected by changes in illumination or other image noise, since they are not learning-based, which in turn causes drifting in the 3D model.

Other methods are based on statistical Point Distribution Models (PDM). Given that human faces have a distinct structure and shape, researchers have developed models and algorithms that exploit this prior information. A popular example is the Active Shape Model (ASM) [19, 20, 30] and its "cousin", the Active Appearance Model (AAM) [31]. Since their introduction, there have been numerous extensions to them (e.g., [78]). ASM is a linear generative model based on learning a statistical model of shape variation and it is also discriminative in the sense that is utilizes local texture information together with a trained texture prior. They work generally well for faces and expressions that match their training data, but can fail badly, if they encounter an expression not in the training set or if they get stuck to a local minimum due to bad initialization.

The major limitation of ASM is that it is a linear model, hence it cannot model the non-linearity of the facial shape manifold across different poses [80]. For example, during a head rotation to a profile pose, some of the facial features can become occluded causing a drastic change in the facial shape, which means the 2D image and 3D face correspondences change as well as the local texture information [57]. A number of extensions have been proposed to overcome this problem, which essentially model this non-linearity with multiple linear ASM models, such as the View-Based ASM [21] and the more successful piecewise linear approximation of Kanaujia et al. [57], which extends ideas in [8, 49]. The latter can handle large out-of-plane head rotations and it can run in real-time by incorporating the KLT feature tracker [100], but it can still fail, especially when some facial component (e.g., eyebrows) gets occluded. This is because the tracker is deterministic and on failure it may remain stuck in a local minimum.

This ASM face tracker would therefore not perform well on certain realistic video, such as sign language video where there are frequent facial occlusions. On the other hand, probabilistic trackers [51] can handle multiple hypothesis to escape local minima.

Therefore we present and validate the hypothesis that tight integration of a face tracker with an adaptive dynamical model (see Sec. 3) improves face tracking accuracy significantly for challenging situations, e.g., in the case of facial component occlusion.

1.1.2 Non-Manual Markers in American Sign Language

Speech recognition technologies have become standard components of modern operating systems, allowing average users to interact with computers verbally. Modern computer systems can interpret voice commands in real time and they can also translate speech to text and vice versa. Recently, we have seen the introduction of smart-phones and tablets and with them came more advanced speech recognition capabilities. These range from the ability to execute internet queries using one's voice, to almost hands-free interaction with your new personal assistant (who now resides inside your phone's software) to schedule meetings, read email, fetch the news, make phone calls, play music, find directions to the nearest gas station, etc. Through such advanced technologies, users can accomplish many computer tasks with minimal typing, making Human Computer Interaction (HCI) an easier and more efficient experience.

On the other hand, technology for the recognition of sign language, which is widely used by the Deaf, is not nearly as well-developed, despite its many potential benefits [75, 76, 77, 84, 115]. First of all, technology that automatically translates between signed and written or spoken language would facilitate communication between signers and non-signers by bridging the language gap. Secondly, such technology could be used to translate sign language into computer commands, hence opening the road for the development of additional assistive technologies (in a manner analogous to existing speech recognition technologies described above) [115]. Moreover, computerized sign language recognition could facilitate the efficient archiving and retrieval of video-based sign language communication [115]. It could assist with the tedious and time-consuming task of annotating sign language video data for purposes of linguistic and computer

Figure 1.2: Illustration of the simultaneous nature of the manual (hand signs) and non-manual (facial expressions and gestures) component of American Sign Language.

science research. Ultimately, such research – and resulting advances in sign language recognition and generation – will have applications that could profoundly change the lives of deaf people and improve communication between deaf and hearing individuals. Non-speaking, non-deaf users of sign language, including some people with autism, aphasia, cerebral palsy, Down Syndrome, and tracheotomies, will benefit from these technologies in the same ways.

However, sign language recognition poses many challenges. First, many of the linguistic components of a sign that must be recognized occur simultaneously rather than sequentially (see Fig. 1.2). For example, one or both hands may be involved in the signing, and these may assume various hand shapes, orientations, and types of movement in different locations. At the same time, facial expression may also be involved in distinguishing signs, further complicating the recognition task. Secondly, there is variation in production of a given sign, even by a single signer. In fact, facial expressions, which constitute the non-manual component of ASL, convey the grammatical information which is crucial to correctly parse ASL sentences (see Sec. 4.1). Additional variation is introduced by the effect of co-articulation, meaning that the articulation of a sign is influenced by preceding and following signs, and by movement transitions between signs (sometimes referred to as "movement epenthesis"). In spite of these challenges, many methods [3, 10, 18, 117, 118, 124] have shown promising results in recognizing manual components of signs.

In sign language, and in ASL in particular, critical grammatical information is expressed through head gestures, such as periodic nods and shakes, and facial expressions such as raised or lowered eyebrows, eye aperture, nose wrinkles, tensing of the cheeks,

and mouth expressions [67, 83]. These linguistically significant non-manual expressions include grammatical markings that extend over phrases to mark syntactic scope (e.g., of negation and questions). Sign language recognition cannot be successful unless these signals are also correctly detected and identified. For example, the sequence of signs "JOHN BUY HOUSE" could be interpreted, depending on the non-manual markings that accompany the signs, to mean any of the following: (i) "John bought the house." (ii) "John did not buy the house." (iii) "Did John buy the house?" (iv) "Did John not buy the house?" (v) "If John buys the house...". In addition, recognition of such grammatical signals can assist with the task of recognizing the manual components of signs. This is because there may be some correlations between information that is expressed manually and non-manually (e.g., [60]).

Motivated by the grammatical importance of head gestures and facial expressions, we present novel methods [75, 76, 77, 84], based on our extended face tracker, for robustly tracking and recognizing such non-manual markings associated with negative sentences, wh-questions, conditional/when clauses, yes/no questions, rhetorical questions and topics. We hypothesize that our extended face tracker allows the extraction of features, which, when coupled with the proposed models, can accurately discriminative the different non-manual markers (see Sec. 4).

1.1.3 Deception Detection

Whether we want to believe it or not, deception has found its way into our everyday social and professional lives [34]. In today's fast-paced, modern world of complex human interactions it is becoming ever more critical to accurately distinguish lies from truth, and friendly from hostile intent. Accurate deception detection can be beneficial not only at an individual level but also on a more global social scale. For example, besides allowing employers to promote the right employee and spouses to catch a cheating partner early in the act, accurate deception detection can aid law enforcement agencies in solving crimes quickly and in saving lives. In addition, it can assist border control and security checkpoint agents to detect potentially dangerous individuals during routine screening interviews [131].

Observer group	μ (%)	σ	Group size	Female (%)	Age μ	Age σ	Exp. μ	Exp. σ
Secret Service	**64.12**	**14.80**	**34**	**3**	**34.79**	**5.96**	**9.12**	**6.69**
Federal Polygraphers	55.67	13.32	60	8	39.42	6.76	6.54	6.19
Robbery Investigators	55.79	14.93	126	2	39.21	8.26	14.77	7.15
Judges	56.73	14.72	110	11	52.64	9.37	11.50	7.77
Psychiatrists	57.61	14.57	67	3	54.24	10.28	23.63	10.28
Special Interest	55.34	15.82	73	53	43.33	13.44	10.76	9.89
College Students	52.82	17.31	39	64	19.90	1.74	N/A	N/A

Table 1.1: Deception detection accuracy and composition of the various groups used in the study of Ekman and O'Sullivan [37]. "Exp." stands for number of years of job experience. Note that even trained and experienced professionals such as secret service agents do not exceed an accuracy rate of 65%.

In response to these potential social and professional benefits of accurate deception detection, many researchers have focused on studying behavior in human interactions, in an attempt to understand the dynamics of deceit and to discover deceptive patterns and cues, if any. This knowledge would then form the basis in designing automatic deception detection systems and for training others to become experts in deception detection [43]. Unfortunately, unaided humans are poor "lie detectors". In fact, in a recent study by Bond and DePaulo [6], it was found that unaided humans can accurately detect deception only 54% of the time and that they are better at detecting audible lies than visible ones. In another study by Ekman and O'Sullivan [37], it was found that various groups of unaided professionals (e.g., judges, federal polygraphers, psychiatrists, etc.) were only 56% accurate (on average) at detecting deception, while even a group of trained and experienced secret service agents achieved an accuracy of 64% (see Table 1.1). The results of these studies should come as no surprise, given the many different verbal and audible cues available during complex human interactions and which an unaided human would need to process simultaneously and in detail, in order to correctly detect deception [16].

It was the poor deception detection abilities of humans, coupled with the demand for fast and accurate deception detection, which led scientists to research development of automated methods and tools to tackle this problem. Currently, the polygraph is the widespread tool of choice, which monitors in real-time changes in heart rate and electro-dermal response from deceit arousal. However, the polygraph has numerous drawbacks [74]. First of all, in order for it to take the necessary measurements it needs to be continuously (and often uncomfortably) connected to the subject being monitored, thus it requires a fully cooperating subject and in close range. The polygraph functions on the premise that under deceit arousal a subject's heart rate and skin conductance may increase. This means that it requires accurate calibration at the beginning of the session by means of a few control questions, in order to establish baseline measurements which will be used later in the interview to detect deception. Nevertheless, sometimes it may fail to give a conclusive reading in spite of a good calibration, as the subject's heart rate may increase for medical reasons for example, which are unrelated to deception.

Furthermore, the polygraph is an overt system. This means that the subject knows they are being monitored and also knows exactly what measurements are being made. As a result, they may devise techniques to trick the machine. For example, they may try to remain calm by taking some drug prior to the exam, in an attempt to relax and maintain a low heart rate. Additionally, they may secretly inflict pain on themselves, both during the calibration phase and later during their truthful responses, so that any excitement that the polygraph registers during their deceptive responses, the polygraph's operator will mistakenly regard as a truthful response.

Lastly, the polygraph requires a trained operator, whose skills and abilities control both the likelihood of human error in the interview and the length of the interview itself. Unlike computers, humans will get tired and will eventually need a break. Therefore, what is needed is an automatic and covert system, which can continuously and unobtrusively detect deception, without requiring the subject's cooperation or any expensive and impractical equipment.

In this thesis, we present methods [16, 74] that can be used to detect deception in an interview scenario using only visual cues, as can be obtained from an un-calibrated

standard camera, in order to distinguish deceptive behavioral patterns. In other words, we hypothesize that an accurate face tracking system, such as the one we present in Chap. 3, can be successfully applied (and extended) to extract, learn and detect, deceptive cues via kinesic analysis of interview video data (see Chap. 5).

1.2 Contributions

Our objective in this work was to build an automatic face tracking system which can be trained to learn the statistical distribution of the shape and texture of human faces across different head poses. In this way, it would be able to track any target face, not necessarily restricted to only those used for training, and adapt to out-of-plane head rotations dynamically, while also gracefully handling occlusion of facial components (e.g., eyebrows). The face tracker should output not just an approximate bounding box for the tracked face, but rather an accurate 2D image localization of specific facial landmarks, corresponding to well-defined facial structures (i.e., nose, eyes, eyebrows, mouth and face outline contour), in addition to a prediction of the 3D orientation of the tracked face. The tracker's output can then be used (directly or indirectly) to recognize dynamic events of facial expressions and head gestures, thus forming the foundation block of useful recognition applications. In particular, we focused on the application of our face tracking system to two very important recognition problems:

- Isolated and continuous recognition of the non-manual markers (involving combinations of head gestures and facial expressions) in video sequences of American Sign Language (ASL). Non-manual markers are important because they convey grammatical information necessary to disambiguate a parsed signed sentence.

- Automatic deception detection using kinesic analysis of visual cues extracted during interview scenarios. Here we first focused on features extracted from the subject's behavioral patterns and then extended it to synchrony analysis of both the subject and their interviewer.

More specifically, the main contributions of this thesis are the following:

- We extend the face tracker of Kanaujia et al. [57] to a probabilistic tracker using a modification of the popular Condensation algorithm [51]. Our novel particle filter uses a hierarchical observation likelihood model that combines anthropometric constraints with hierarchical appearance constraints at both the landmark level and at the facial component level, while also modelling shape cluster transitions during head rotations (inspired from work in [49]). We empirically demonstrate the superiority of our enhanced tracker, relative to the original version [57], on one of the many challenging video sequences present in our ASL video dataset, where the signer's eyebrows get occluded for a relatively long period. Accurate tracking of facial components, especially eyebrows, is crucial for correct recognition of non-manual markers in ASL (see Chap. 4).

- We present a framework for isolated recognition of non-manual markers in segmented ASL video sequences using a "bag-of-words" model built from appearance and head descriptors which are extracted with the help of our face tracker. We empirically show that our method can successfully recognize non-manual markers associated with wh-questions and negation [75, 84]. Our method can be extended to use any set or sets of appearance features, instead of or in addition to the ones we present, as well as extended to recognition of facial expressions and head gestures in other domains.

- We extend our framework for isolated recognition in segmented video sequences to recognize additional classes of non-manual markers, some of which appear similar when viewed in a static context, namely yes-no questions, conditional/when clauses and topics, in addition to wh-questions and negative sentences. We achieve this by introducing a rich feature descriptor, which we refer to as "oracle features" and encodes feature dynamics, and the idea of Multiple Instance Feature (MIF) [68] for handling feature misalignment. Additionally, we use the discriminative Hidden Markov Support Vector Machine (HMSVM) [1] for modelling temporal relationships between neighboring frames, instead of the static Support Vector Machine (SVM) [13], which we had previously used. We validate the positive

effect that each proposed extension (oracle features, MIF, HMSVM) has on the overall accuracy of the framework with experimental results [77].

- The isolated recognition of non-manual markers in segmented ASL videos is not tremendously useful in practice, simply because it assumes that we have already acquired in some way the start and end of each non-manual marker that we want to recognize, e.g., using a spotting algorithm. Continuous recognition is more practical because it bypasses this requirement, thus saving computational resources. Therefore, our next contribution is the extension of our framework to continuous recognition using an extended feature set (encoding both texture and appearance of the region of interest) and spectral clustering [4, 85] to learn the appearance manifold of the facial expressions associated with the non-manual markers of wh-questions, negative sentences and topics [76].

- Head pose variation changes the appearance of the region of interest, adding additional complexity to the task of recognizing non-manual markers in ASL, especially for classes that are only subtly different. For example, topics and yes-no questions both involve raised eyebrows, but with the latter the head is jutted forward, while with topics it may be tilted back. Small training datasets may not exhibit all possible variations in the production of these non-manual markers, therefore resulting in inaccurate recognition models. Our next contribution is the introduction of a method for 2D image warping of non-frontal poses to frontal, using a trained 3D model [5, 128]. We empirically show that application of this warping transformation to the region of interest, prior to feature extraction, filters out the effects of head orientation on the appearance of the face region. It also removes the effects of foreshortening on the estimation of eyebrow height, which is critical for detecting non-manual markers involving raised or lowered eyebrows.

- In the area of automatic deception detection, our contribution is a non-parametric log-feature space feature representation of features, which are extracted using the face tracker and a skin blob tracker (e.g., [70]), and subject-specific modelling of behavioral profiles, for automatically discriminating over-control, agitated and

relaxed states. We show that our method [74, 16] for kinesic analysis of a subject's behavioral patterns in an interview scenario outperforms state of the art accuracy of similar approaches (e.g., [73]).

- Lastly, there have been a number of studies that point to a correlation of the degree of synchrony between a subject and their interviewer and the deceptive state of the subject (e.g., [81]), yet there has not been much work in developing a computerized method for it. We extend our methods for deception detection through kinesic analysis of a subject's behavior, to deception detection through synchrony analysis of the subject and their interviewer. We present a feature representation which quantifies the level of synchrony and show proof-of-concept experimental results, which validate our approach.

1.3 Organization

The remainder of this thesis is organized as follows. Chapter 2 provides a review of relevant work in face tracking (Sec. 2.1), in computerized recognition of manual and non-manual markers in American Sign Language (Sec. 2.2) and in automatic deception detection (Sec. 2.3). In the review we point out the limitations of the existing methods, thus clarifying our contributions.

Chapter 3 describes our proposed method for probabilistic face tracking, which uses a modified particle filter with anthropometric and appearance constraints, both at the landmark level and at the facial component level. We begin with an overview of Active Shape Models (Sec. 3.1) and then introduce our hierarchical particle filter extension (Secs. 3.2–3.5). We conclude the chapter with experimental results (Sec. 3.6), where we illustrate that our extended face tracker can handle temporary occlusions of entire facial components, which is crucial, for example, for the accurate tracking of facial features (e.g., eyebrows) that have a linguistic significance in Sign Language.

Chapter 4 describes the application of our extended face tracker (Chap. 3) to the recognition of non-manual grammatical markers in video sequences of American Sign Language (ASL). We begin the chapter with the relevant linguistic background on

non-manual grammatical markers of ASL (Sec. 4.1). Next, we present our framework for recognition of segmented sequences (Secs. 4.2–4.3), meaning sequences where we have isolated the segments containing some non-manual marker. This is followed by a description of our framework for continuous recognition of unsegmented sequences (Secs. 4.4), meaning sequences in which we do not assume we know when some non-manual marker occurs, and of our technique for 2D image warping based on a 3D face model to achieve normalization of our input features (Sec. 4.5). We include various experimental results for both isolated and continuous recognition (Secs. 4.2.4 and 4.4.4), which demonstrate the effectiveness of our methods.

Chapter 5 describes the application of our extended face tracker (Chap. 3) to automatic deception detection from visual cues only. We begin the chapter with some relevant theoretical background (Sec. 5.1) to justify our approach and present our hypothesis. Next, we present our methods for target tracking (Sec. 5.2), followed by a description of our feature extraction stage (Sec. 5.3). Moreover, we introduce our novel non-parametric descriptor (Sec. 5.3.3) for representing behavioral patterns in a way that helps distinguish over-controlled and agitated behavioral profiles from subjects' normal baseline patterns, as well as the model used for recognition (Sec. 5.4). We conclude the chapter with an extension to the system that relies on interviewer-subject synchrony (Sec. 5.5) and with experimental results (Sec. 5.5.3) to validate our methods.

Finally, in Chapter 6 we summarize our methods and findings. We conclude with a discussion of possible directions for future work that could be used to extend our system.

Chapter 2

Overview of Related Work

In this chapter we present an overview of work related to the proposed face tracking system and to each of the presented applications of it. Namely we review work related to recognition of the non-manual component of ASL and to deception detection from kinesic analysis using only visual cues. Given the popularity of both the topic of face tracking and the presented applications, we focus on work most relevant to what is presented herein.

2.1 Face Tracking

Deformable 3D models have been widely used for face tracking as well as for facial animation [5, 22, 23, 62]. A few extensions to these parametrized 3D deformable models include outlier rejection [116], use of Kalman Filters [44], as well as incorporation of optical flow constraints as an additional cue to the feature pool [24]. However, update of the model parameters still depends on the accuracy of the extracted image features. Given that the methods for extracting this features are in some cases deterministic or not learning-based, they can drift if their assumptions are violated, e.g. due to illumination changes, causing the 3D model to also drift. Moreover, Kalman filters behave well for linear systems but for non-linear systems, particle filter implementations, e.g., Condensation algorithm, [49, 51] achieve better performance and additionally, observation likelihood models can be custom tailored and even combined, in a more natural way, to suit the given application.

Other methods are based on statistical Point Distribution Models (PDM). Given that human faces have a distinct structure and shape, researchers have developed models and algorithms that exploit this prior information. A popular example is the Active

Shape Model (ASM) [19, 20, 30] and the closely related Active Appearance Model (AAM) [31]. ASM-based methods build statistical models of the shape of the object they want to detect and track (e.g., faces) from a set of training images, which have been labelled with the 2D or 3D coordinates of predefined landmarks, characteristic of the object's shape and structure. In the case of face tracking, these landmarks can be on the face's contour, around the eyes, eyebrows, nose and mouth. The shapes from the labelled training images are aligned using Procrustes analysis [46] and through the application of Principal Component Analysis (PCA) the model learns the permissible modes of shape variation and the texture profiles that the model should expect to find around each landmark in a test image. The AAM on the other hand can model texture variations on the entire face region, so it gives a better match to the texture of the test face but its running time is slower than that of the ASM. Both models, however, are sensitive to their initialization and can get stuck in local minima.

In its original formulation, the ASM assumed that the errors between the model fit and the test image are distributed normally [20]. There have been numerous extensions to the classical ASM since. In the work of Cootes et al. [19] they use a mixture of Gaussians to model shapes, while Romdhani et al. [96] introduced Kernel PCA in an attempt to overcome the linearity limitation of the original formulation. Kernel PCA allowed them to model non-linear shape variation resulting from changes in the yaw head angle. Other extensions include the work by Milborrow et al. [78], Li and Ito's Adaboost-based ASM [66], the work by Rogers and Graham [95] and the work of Jiao et al. [52], who incorporated wavelets into the face alignment algorithm.

The major limitation of ASM is that it is a linear model, hence it cannot model the non-linearity of the facial shape manifold across different head poses [80]. A number of extensions have been proposed to overcome this problem, such as the View-Based ASM [21], which requires explicit modelling of head pose angles, and the more successful piecewise linear approximation of Kanaujia et al. [57], which extends ideas in [8, 49]. The latter can handle large out-of-plane head rotations but it can still fail, especially when some facial component (e.g., eyebrows) gets occluded. This is because the tracker is deterministic and on failure it can get stuck in a local minimum. Gong et al. propose

Figure 2.1: Comparison of face tracking methods under occlusion: (Left) Tracking result from our proposed extended face tracker. Eyebrow occlusion by signer's hair is properly handled; (Right) Result on the same frame using Yang et al.'s [127] sparse shape registration method, which fails to estimate the correct position of the occluded eyebrows, mainly because it does not utilize dynamic information.

using non-linear projections onto the eigen-space [45], while Zhou et al. [134] propose a Bayesian multi-view model but their reliance on the EM algorithm to estimate shape parameters renders their implementation impractical for real-time applications.

Another shortcoming of ASM models is their inability to handle occlusions. If a part of the object is occluded the landmarks corresponding to the occluded parts cannot find a good match. Even worse, in their effort to reach a local optimum, they may even cause a drift in the landmarks that did manage to find an accurate match, resulting in a poor overall registration of the target. In order to address this issue, Felzenszwalb et al. [39] and Tian et al. [105] propose pictorial structures to model the spatial relationships between the different components of the target object, e.g., in the case of human tracking these components would be the head, torso, arms and legs and the modelled relationships would be their connectivity as well as joint angle limits. In [48] the EM algorithm is used with a generative model, while Zhou et al. [133] propose a tangent shape approximation. Saragih et al. [97] introduce regularized landmark mean-shift but their optimization is only at the landmark level, while our proposed extension fuses multi-level appearance information as well as anthropometric constraints. Recently, Yang et al. [127] used sparsity to model the error term in model fitting under occlusion. Although promising, this method does not incorporate

dynamic information and only looks at a given frame in a static context. Therefore, it fails to correctly track faces in our challenging sign language videos, where motion information is critical in order to estimate the position of the occluded landmarks, while the extended face tracking method we propose succeeds (see comparison in Fig. 2.1).

2.2 Sign Language Recognition

Most research on computer-based sign language recognition has focused on the manual components of signs. A thorough review of early such efforts is presented in the survey by Pavlovic et al. [92]. More specifically, Starner and Pentland [102] use color tracking and HMMs to recognize a 40 word lexicon of manual signs. In the work of Vogler and Metaxas [118], the manual signs are split into independent movement and hand shape channels, and an HMM framework is used to model signs as a sequence of phonemes. These independent channels allow them to handle simultaneous manual events. Bauer and Kraiss break down signs into smaller units using unsupervised clustering, achieving high recognition accuracy in isolated sign recognition experiments [3]. In [120], the authors apply techniques from speech recognition to develop a method that quickly adapts to unknown signers, in an attempt to handle interpersonal variance. Similarly, the authors of [136] present a method for sign recognition, which uses a background model to achieve accurate feature extraction and then performs feature normalization to achieve person independence. To tackle the problem of self occlusions of the hands, Martinez and Ding [27] first perform 3D hand reconstruction and then represent hand motions as 3D trajectories. Lately, we have even seen the emergence of some weakly supervised methods that successfully attempt to learn manual signs from TV subtitles [10, 18].

Recently researchers have begun to address the importance of facial expressions for sign recognition systems [91]. Von Agris et. al. [121] provide an extensive review of recent developments in visual sign recognition, together with a system that uses both the manual and the non-manual components of signs (see Fig. 2.2). However, their system poses the restriction that the signer must be wearing a glove with colored markers, in order to enable robust hand tracking and hand posture reconstruction. Additionally,

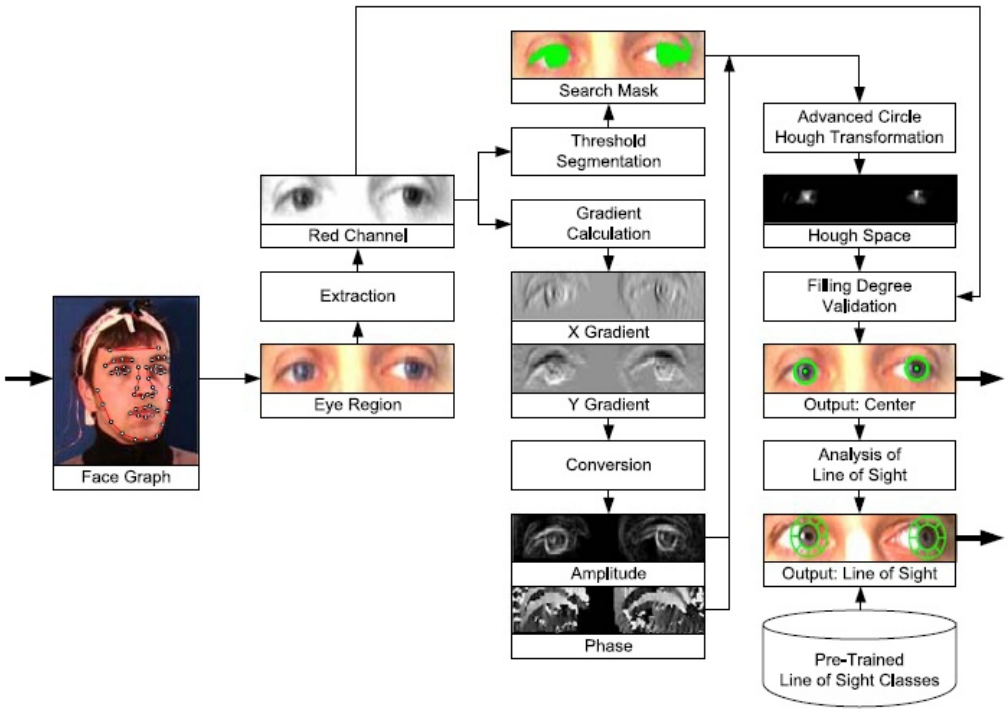

Figure 2.2: Schematic representation of the procedure proposed in [121] for identifying line of sight of a subject. The procedure is a component of the overall system described therein for visual recognition of sign language.

in their system, the tracked facial features are not used to recognize facial expressions which have grammatical meaning. Vogler and Goldenstein present a 3D deformable model for face tracking, which emphasizes outlier rejection and occlusion handling [114, 115] at the expense of slower run time. They use their system to demonstrate the potential of face tracking for the analysis of facial expressions encountered in sign language, but they do not use it for any actual recognition of facial expressions (see Fig. 2.3). Similarly, in [87] optical flow tracking with probabilistic subspaces is utilized to handle occlusions. Both models have many parameters, thus are difficult to train without over-fitting. In our work, we use an Active Shape Model (ASM) face tracker [20, 57], which can better handle out of plane rotations by modelling the facial shape manifold as multiple overlapping clusters, while being able to run in real-time in a single-threaded environment and no GPU optimizations. Additionally, we use a particle filter extension with a hierarchical observation function for probabilistic tracking with better handling of partial occlusions (see Sec. 3).

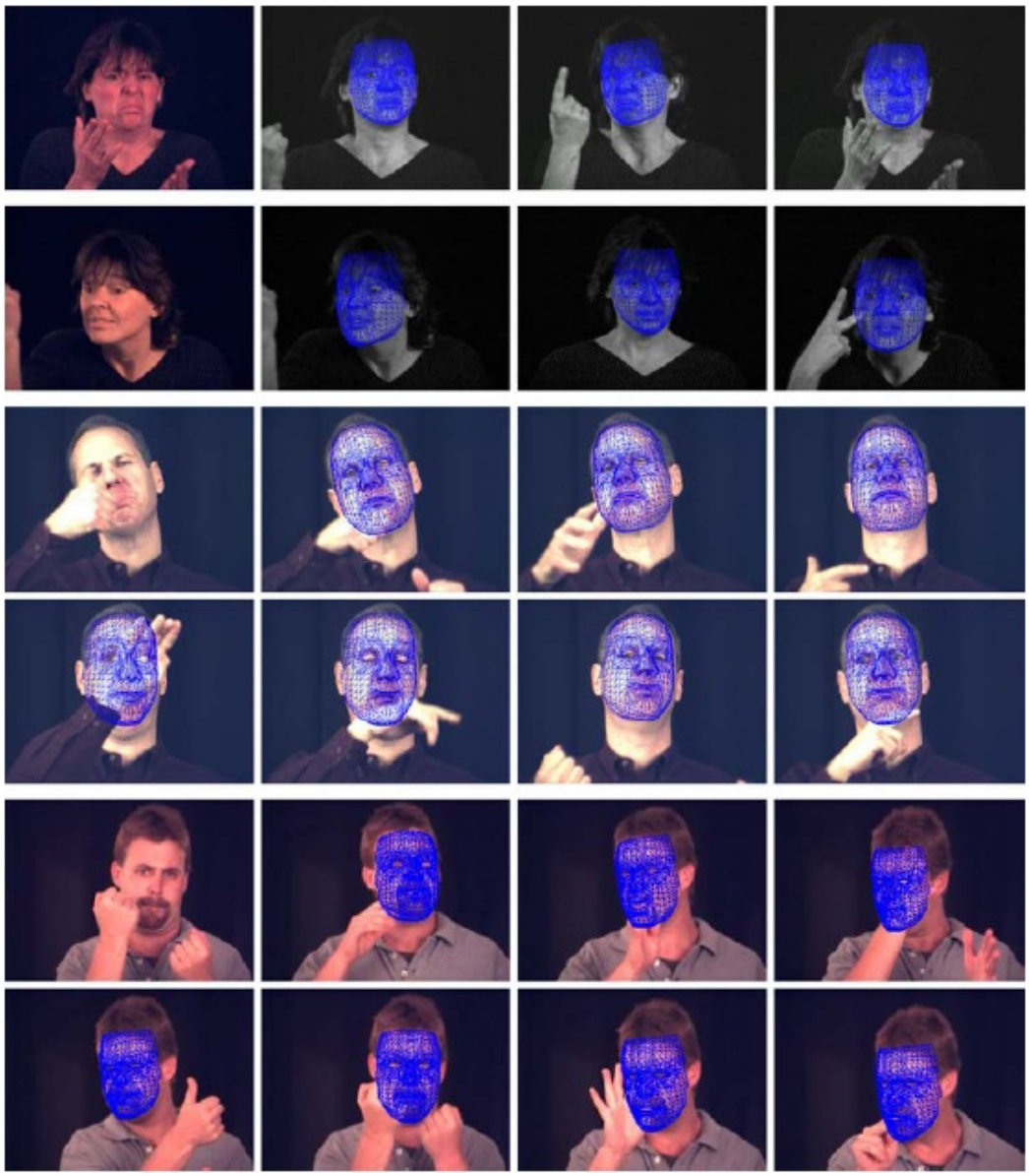

Figure 2.3: Sample tracked frames produced by the framework of Vogler and Goldenstein [114].

Following the initial interest in the non-manual component of signed languages, researchers have begun designing multi-modal recognition frameworks fusing manual and non-manual features [2, 59, 60, 98, 119]. However, they limit themselves to recognizing head gestures of negation (because their focus is still the manual component and how to improve recognition performance by utilizing non-manual information e.g., [59]) and eyebrow movements (up or down) [98], instead of differentiating the overlapping classes of non-manual markers (e.g., topics and yes-no questions). Additionally, they do not explicitly handle occlusions of facial components, and their tracking is deterministic and unable to handle large and fast 3D head rotations.

More specifically, Aran et al. [2] present a system for sequentially fusing manual signs and non-manual gestures, in the form of head motions, by first identifying the level of uncertainty of a classification decision, identifying sign clusters, and identifying the correct sign based on the manual sign and head motion. Similarly, Kelly et al. [60] present a multi-modal system for continuous recognition of Irish Sign Language but also focus on fusing head motions with hand gestures for the purpose of improving recognition of the manual signs and facial expressions are not recognized. Recognition is done using multichannel HMM threshold models using continuous multidimensional observations. Von Agris et al. [119] recognize manual signs by integrating manual features extracted using multiple hypothesis color tracking, and facial features, which serve to encode facial expressions, extracted using an Active Appearance Model (AAM) [31] to identify facial components (e.g., eyes, mouth) from which geometric features are computed. Recognition of manual signs is done by Hidden Markov Models and feature-level fusion of manual and non-manual features. Similarly, in [129] the neutral and the six universal facial expressions, (i.e., anger, disgust, fear, happiness, sadness, surprise) plus a facial expression of question are recognized and used as additional information to help with recognition of the manual component in a Conditional Random Field [103]. An AAM is used for face tracking and classification is via Support Vector Machines (SVM) [13].

As far as head gestures are concerned, Erdem and Sclaroff [38] present a 3D head tracker to detect head gestures relevant to ASL, e.g., head shakes and head nods. The

method relies on the peaks and valleys of the head motion signal, therefore it does not need training but it assumes that there are no head occlusions and no appearance changes after its initialization. The system developed by Kelly et al. [61] achieves a similar task using a feature extraction scheme based on a cascade of boosted classifiers for face and eye detection. Head gestures are recognized by an HMM network. Similarly, Xu et al. [126] deal with recognition of various kinds of head motion in Japanese Sign Language, which often occurs at the break between words or at the boundaries of a sentence, thus can provide grammatical constraints for JSL segmentation. Their feature extraction is based on color tracking and image moments to detect the face and its orientation respectively. Lastly, Ding and Martinez [26] introduce a method for detailed detection of faces and facial features which is based on learning to discriminate between features and their surroundings, and on a voting strategy over different scales. They then localize facial features using image gradient and color information.

Finally, we review work focused only on recognition of non-manual markers in sign language. Ming et al. [79] utilize Gabor wavelet networks and Independent Component Analysis [17] to recognize upper face non-manual signals like yes-no questions, negative yes-no questions and wh-questions, as well as lower face non-manual signals, such as pursed lips, but neither of their method uses the 3D head pose and head motion information. These are also computationally intensive and fail to distinguish certain non-manual signs that appear similar. Nguyen and Ranganath [86] began their efforts toward recognition of non-manual markers in sign language with a face tracking system based on the Kanade Lucas Tomasi (KLT) feature tracker [100]. However, KLT is prone to drifting, so in order to cope with large head motions and occlusions and to adapt to face shapes of different people, they also use a Bayesian feedback mechanism which incorporates Probabilistic Principal Component Analysis (PPCA) [107] and an on-line update mechanism. In [87] they extract facial features using this tracker and train Hidden Markov Models to recognize facial expressions and gestures, e.g. head shake, eyebrow raise. The probabilities output by HMM are then fed to a Neural Network to recognize yes-no questions, wh-questions, topics and negation. In their latest work [88] they use a similar tracking and feature extraction framework, but for recognition they

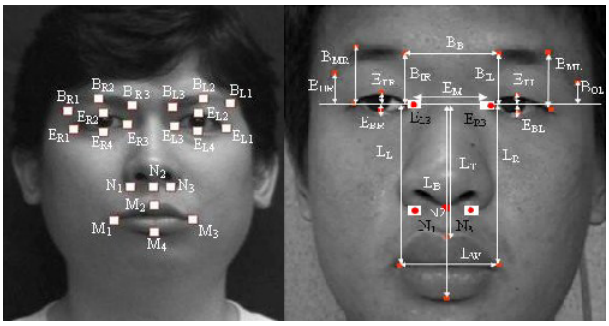

Figure 2.4: Illustration of the tracked feature points (left image) and the geometric features extracted from them (right image) and used by Nguyen and Ranganath [88].

use a hierarchical CRF model [103], which recognizes facial expressions and gestures at the bottom level and at the top level fuses the likelihoods to recognize the non-manual markers. Still, their system relies heavily on the tracking result given it only uses geometric features (see Fig. 2.4), while in our methods we incorporate appearance features as well [75, 76, 77, 84]. Moreover, we present a method for feature normalization in terms of head pose, by warping input images to a frontal pose before extracting our features. In this way, we filter out the effects of foreshortening of geometric features such as the eyebrow heigh when the head is pitched up or down. This also filters out the effects of head pose on the appearance features, while we extract head pose information explicitly (see Sec. 4.5).

2.3 Deception Detection

Related work in automatic deception detection can be categorized into two groups. In the first group, researchers look for physiological indicators, which they can correlate to deception, in a similar fashion to how the polygraph works, which correlates elevated heart rate and skin conductance to deceit arousal [122]. For example, the authors of [9] and [110] build a thermodynamical model to monitor increases in blood flow in the region between the eyes (periorbital region) of a subject, using thermal cameras (see Fig. 2.5 for an illustration). However, this method needs a controlled environment and expensive imagery equipment, thus hindering its broad deployment. Moreover, the method in [9] cannot track head movements and its accuracy suffers, if the subject's

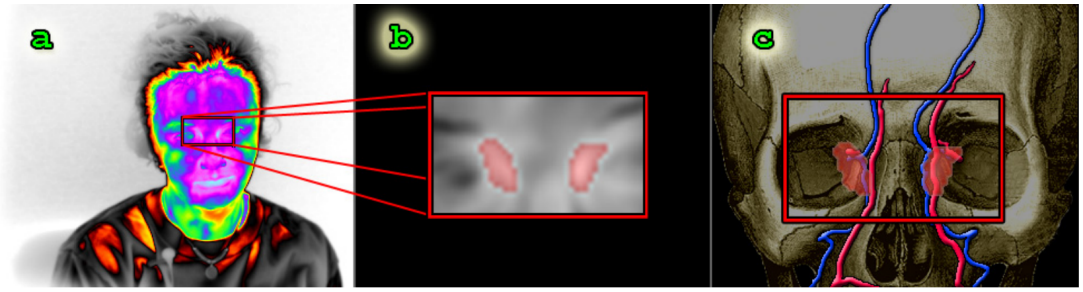

Figure 2.5: Illustration of the periorbital region analyzed by methods that rely on physiological indicators to detect deceit [110] by correlating it to increased temperatures in this region; (a) Thermal input image with the periorbital region of interest marked by the red rectangle (b) Blown-up periorbital region of interest where the hottest 10% pixels are shown in pink (c) The periorbital region of interest in (b) superimposed on tan image of the facial and ophthalmic arteriovenous complex.

head is moving or is at an angle to the camera. While in [110] the authors introduced a tandem Condensation tracker for better head tracking, whereby they track a more trackable region and use its estimated position to predict the location of the periorbital region, the method's accuracy can still be affected by fluctuations in background temperature which can distract the tracker. Similarly, some researchers, such as the authors of [41, 54, 63], use functional Magnetic Resonance Imaging (fMRI) to monitor brain activity during interviews and to detect deception based on which areas of the brain are activated (see Fig. 2.6 for an illustration of fMRI brain images and consistent areas of activation during deceit arousal). However, methods based on fMRI cannot be used in a covert scenario, they require specialized, expensive and sensitive equipment in a shielded environment, and a cooperative subject.

In the second group, researchers move away from physiology and attempt to analyze behavioral indicators of deception, instead. Zhang et al. [131] examine which facial Action Units (AU) are activated in a particular facial expression, in order to determine whether it is faked or real. The method is based on the fact that for specific facial expressions (e.g., smiling) there is a predefined set of involuntary Action Units which are activated [34, 36] and which the vast majority of the population cannot voluntarily control (e.g., in the case of smiling it is the muscles around the eyes). If these are not detected then the expression is deemed to be fake. Their method, however, is currently based on static images and cannot readily be used to detect, for example, deceptive

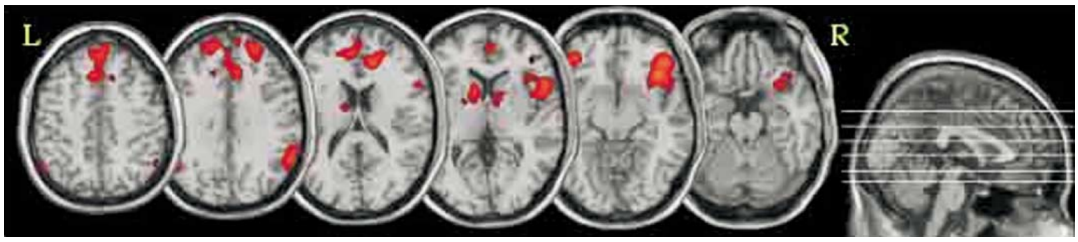

Figure 2.6: Illustration of fMRI brain images analyzed by methods such as [63], which attempt to discover correlations between deception and the human neural system. Regions of consistent activation during deception are marked in red.

responses to interview questions in video streams; only faked facial expressions. Lu et al. [70] use color analysis to track hand and head skin blobs of subjects (see Fig. 2.7), to classify their movement signatures as over-controlled, relaxed or agitated. However, it is not convincing that the equation they used for state estimation generalizes to unseen data, given they only tested it on a small dataset (five subjects). Given also that there is vast physical and cultural variation in the human population, we believe state thresholds may be subject or culture dependent, so subject-specific modelling may be more appropriate. Tsechpenakis et al. [109] extend the work of [70], translating blob features into illustrator and adaptor behaviors. They combine these via a hierarchical Hidden Markov Model [94] to decide if the subject is agitated, relaxed or over-controlled but also report results on relatively small datasets, so it is still not convincing that there exists a global definition of agitated, relaxed and over-controlled states. In the work of Meservy et al. [72, 73], the step of classifying behaviors [70, 109] is bypassed and the authors attempt to directly derive deceptive cues, using blob analysis as in [70]. They segment the video data of interviews into responses and use summary data of each segment, collapsing it to a mean and variance for each feature (e.g., mean and variance of head blob velocity), to make predictions but they do not achieve high accuracy.

We believe that relying on the parametric representation (mean and variance) of the summary blob data used in [70, 73, 109], causes a lot of useful information about each feature's distribution to be lost, smoothing out any abrupt gesturing motions and expressions that briefly occur, when a subject is being deceitful. Eckman and Friesen name this "leakage" [35], while Buller et al. refer to it as "non-strategic behavior"

Figure 2.7: Illustration of tracked skin blob regions of the head and hands (shown as ellipses) analyzed by methods that compute behavioral indicators through kinesic analysis for the purpose of deception detection [109].

[12]. We propose to extract "motion profiles" (see Sec. 5.3), which differ from the movement signatures of [70, 73, 109], in that ours are non-parametric representations of the distributions of both blob and facial features (see Sec. 5.3.3). In this way, this richer representation captures any such leakage that occurs during an interview response and our method can benefit by incorporating features extracted from the facial region as well. Our results verify a claim made by Burgoon [15], who argued that gross body gestures and animations, facial and hand adaptors, and head gestures may be more reliable than facial expressions, but we also show that inclusion of facial features (e.g., mouth asymmetries, which have been previously used to successfully detect facial expressions associated with stress [28], and eyebrow movements) in the more reliable feature pool causes a small improvement in detection accuracy (see Sec. 5.4).

Chapter 3

A Bayesian Filtered Face Tracker

In this chapter we describe the core of our system, namely the face tracker. It is based on the ASM model proposed by Kanaujia et al. [57], which can handle out-of-plane head rotations and can estimate the head's 3D orientation angle. We extended it to better handle dynamic discontinuous shape changes, e.g., when the head changes pose, and occlusion of facial components. The extension we present is a modification of the Condensation algorithm [51], using particles with a mixed-state (to include shape cluster information) and observation likelihood models that model anthropometric and hierarchical appearance constraints. We qualitatively and quantitatively illustrate the benefits of our extended face tracker.

3.1 Overview of Active Shape Model

The classical ASM model [20] is a statistical model of the permissible modes of variation in the shape of a class of objects. In reality, the ASM model consists of two sub-models: (1) a global shape model, which models shape variation and (2) a model of local profiles for each landmark, which model the texture (in the form of grey-level image gradients) around each landmark.

Suppose we start with m labelled training images and that each image is annotated with n landmarks. The training shapes are aligned using Procrustes analysis [46]. Let the i^{th} shape be represented with the $2n$-dimensional vector $\mathbf{x}_i = [x_{1,x}, x_{1,y}, \ldots, x_{n,x}, x_{n,y}]^T$, where $x_{j,x}$ and $x_{j,y}$ are the (x, y) coordinates of the j^{th} landmark respectively, so that the training set is given by the set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$. Suppose that the covariance of the training set is given by matrix $\mathbf{S}$ and that its eigenvectors, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{2n}]$, span a linear subspace in which valid face shapes reside.

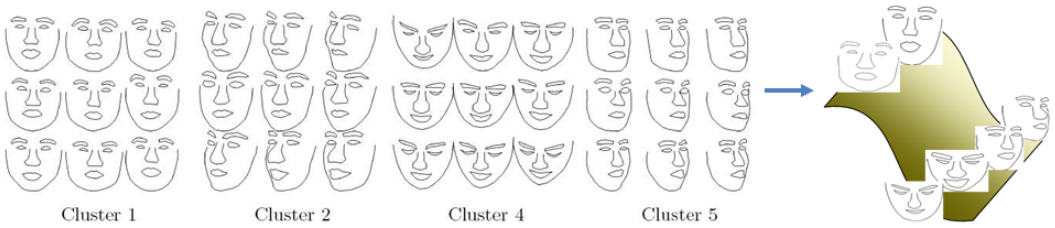

Figure 3.1: Illustration of the approach used by Kanaujia et al. [57] to model the non-linearities of the facial shape manifold. The idea is to collect training facial shapes depicting a variety of head poses. These are clustered by head pose, so that shapes of similar pose end up in the same cluster, learning a separate ASM model for each pose.

Define the corresponding matrix of eigenvalues as $\mathbf{\Lambda}$. $\mathbf{P}$ and the corresponding $\mathbf{\Lambda}$ are calculated by Principal Component Analysis on the shape covariance matrix $\mathbf{X}$. The eigenvectors corresponding to the largest eigenvalues represent major modes of shape variation, so that any new test shape $\mathbf{x}$ can be approximated by the mean shape in the training set, $\bar{\mathbf{x}}$, and a linear combination of the eigenvectors representing the largest shape variance (e.g., 95% of total variance) of the subspace, can be approximated as:

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} = \bar{\mathbf{x}} + \sum_{k=1}^{q} \mathbf{p}_k b \ , \tag{3.1}$$

where $\mathbf{b} = [b_1, b_2, \ldots, b_q]^T$ is the encoding of the test shape, assuming that the q eigenvectors with the largest eigenvalues have been kept after the application of PCA.

Face trackers built on the classical ASM suffer from the fact that changes in viewpoint, resulting from out-of-plane rotations, cause the facial shape to lie on a non-linear manifold, therefore linear methods cannot accurately model it. Inspired by work in [8, 49], Kanaujia et al. [57] presented a generic framework for learning such non-linearities in shape space using a set of overlapping linear subspaces, essentially a collection of ASM models for the different head poses one expects to find in real data (see Fig. 3.1). Their method differs from [21], where multiple independent ASM models are trained and switched during tracking, depending on the head pose, which results in abrupt shape changes and inaccurate fitting. Instead, the shape clusters are allowed to overlap so that during the image search algorithm, the model is allowed to switch

models as needed. The training and image search algorithms follow Alg. 1 and 2 respectively. However, the lack of a stochastic dynamic model to model cluster transitions and shape changes means that this method still suffers from local minima and occlusions. The particle filter extension, which we present next, addresses these issues by modelling:

1. Anthropometric constraints (to prevent abrupt shape changes)
2. Hierarchical appearance constraints (to keep the tracker locked on the target)
3. Cluster transitions (to allow escape from local minima)

The nature of the particle filter (i.e., weighted sample/particle set) also allows the tracker to maintain multiple hypotheses about the target's position, as well as automatic detection of loss of track, reflected in a low sample likelihood, in which case, the system re-samples the state space, until the target is re-acquired.

Algorithm 1 ASM Training Algorithm [57]

---

1. Align the set of training shapes $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ to the mean shape $\bar{\mathbf{x}}$ using Procrustes analysis [46] to get a set of aligned training shapes $\mathbf{X} = \{\mathbf{x}_{1,a}, \mathbf{x}_{2,a}, \ldots, \mathbf{x}_{m,a}\}$.
2. Project each aligned shape $\mathbf{x}_{i,a}$ to the tangent space of the mean shape using $\mathbf{x}^t_{i,a} = \mathbf{x}_{i,a}/(\mathbf{x}_{i,a} \cdot \bar{\mathbf{x}})$.
3. Cluster the resulting shapes to N clusters using the EM algorithm, enforcing a minimum covariance $\mathbf{V}_{\text{floor}}$, which ensures that the clusters are overlapping.
4. After the assignment to clusters, align the shapes to the local mean shape of their cluster c, projecting to the tangent space of the cluster mean to get $\mathbf{x}^t_{i,a,c}$.
5. Train local linear PCA models for each cluster, so that shapes within each cluster c are approximated as $\mathbf{x}^t_{i,a,c} \approx \bar{\mathbf{x}}_\mathbf{c} + \mathbf{P}_c \mathbf{b}_{c,i}$.
6. Learn Gaussian mixture models per cluster for the intensity profiles around each landmark.

Algorithm 2 ASM Image Search Algorithm [57]

1. Assign initial cluster, c, based on Mahalanobis distance to local cluster mean shapes.
2. Search along the normal of each landmark to maximize mixture probability of the intensity profile models for current cluster, c, to get shape $\mathbf{x}$.
3. Get the eigen-encoding, $\mathbf{b}_c$, by projecting on the current cluster's shape subspace, $\mathbf{P}_c$, using: $\mathbf{x} = \bar{\mathbf{x}}_\mathbf{c} + \mathbf{P}_c\mathbf{b}_c$, and truncating the resulting eigen-parameters, $\mathbf{b}_c$.
4. Assign new current cluster, c, based on Mahalanobis distance to local cluster mean shapes (i.e., cluster switching occurs in this step). If cluster switching occurs, re-estimate new $\mathbf{b}_c$.
5. If shape $\mathbf{x}$ has converged then return, else go to step 2.

3.2 Particle Filters: Condensation Algorithm

The Condensation (Conditional Density propagation) algorithm [51] is a flavor of particle filter, which, in turn, is a Bayesian sequential importance sampling technique, whereby the posterior distribution of the state of the system being modelled (in our case, the state of the tracked object, i.e., the face), is modelled by a finite set of weighted samples at time, t: $\mathbf{S}_t = \{(w_t^1, \mathbf{s}_t^1), (w_t^2, \mathbf{s}_t^2), \ldots, (w_t^n, \mathbf{s}_t^n)\}$. A sample (also known as particle, hence the name particle filter) represents a state configuration, $\mathbf{x}$, and the associated weight, w, is a measure of its likelihood, i.e., its importance. For purposes of clarity, in this section we will use the symbol for samples and states interchangeably, since a weighted sample is essentially a weighted state.

Bayesian sequential importance sampling involves two main steps: (1) prediction and (2) update. Let $\mathbf{z}_t$ be the observation about the tracked system and let $\mathbf{x}_t$ be the true system state, both at time t. In the prediction step, given all available observations up to time $t-1$, $\mathbf{z}_{1:t-1} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{t-1}\}$, this technique uses the stochastic dynamic model of the system (also called the probabilistic transition model), $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, to predict the posterior probability of the state at time t, using:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_t|\mathbf{x}_{t-1})d\mathbf{x}_{t-1} \ . \tag{3.2}$$

Once the new observation, $\mathbf{z}_t$, arrives at time t, the system state is updated using Bayes rule:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \quad , \tag{3.3}$$

where $p(\mathbf{z}_t|\mathbf{x}_t)$ is governed by the observation model.

As already mentioned, in the particle filter framework, the posterior is approximated by a finite weighted sample set, $\mathbf{S}$. Candidate samples, $\mathbf{s}_t^i = \mathbf{x}_t^i$ are drawn from an importance distribution, $q(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$, and the sample weight is given by:

$$w_i^t = w_i^{t-1} \frac{p(\mathbf{z}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{z}_{1:t-1})}{q(\mathbf{x}_t^i|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})} \quad . \tag{3.4}$$

For the Condensation algorithm, $q(\mathbf{x}_t^i|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$, so the weights in Eq. 3.4 become proportional to the observation likelihood, $p(\mathbf{z}_t|\mathbf{x}_t)$. At a particular moment in time, we have a weighted particle set of n samples. To move the sample set one step in time, we repeat the following n times. We select a sample from this set, with sampling probability according to its weight. In this way, samples with heavier weights are more likely to be selected and propagated. The selected sample is propagated in time, as per the transition model of the system, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and then some random noise is added to the sample to diffuse the resulting sample set and prevent degeneracy. At this point we have propagated all samples in time and we assign to all the same weight, $w_t^i = 1/n$. When the new observation comes in, $\mathbf{z}_t$ we use it to evaluate the likelihood (weight) of each sample, $\mathbf{s}_t^i$, according to our observation model, $w_t^i = p(\mathbf{z}_t|\mathbf{s}_t^i = \mathbf{x}_t)$. The weights are normalized to unit sum and the result is a propagated weighted sample set. Now we can use the sample mean, or the sample median or even mode, depending on the application, to predict the system's state. The process is then repeated. See Fig. 3.2 for an illustration.

3.3 Stochastic Modelling of Cluster Transitions

Often times the ASM face tracker can get stuck in a local minimum from which it cannot recover. In order to help prevent this, we present our model for probabilistic

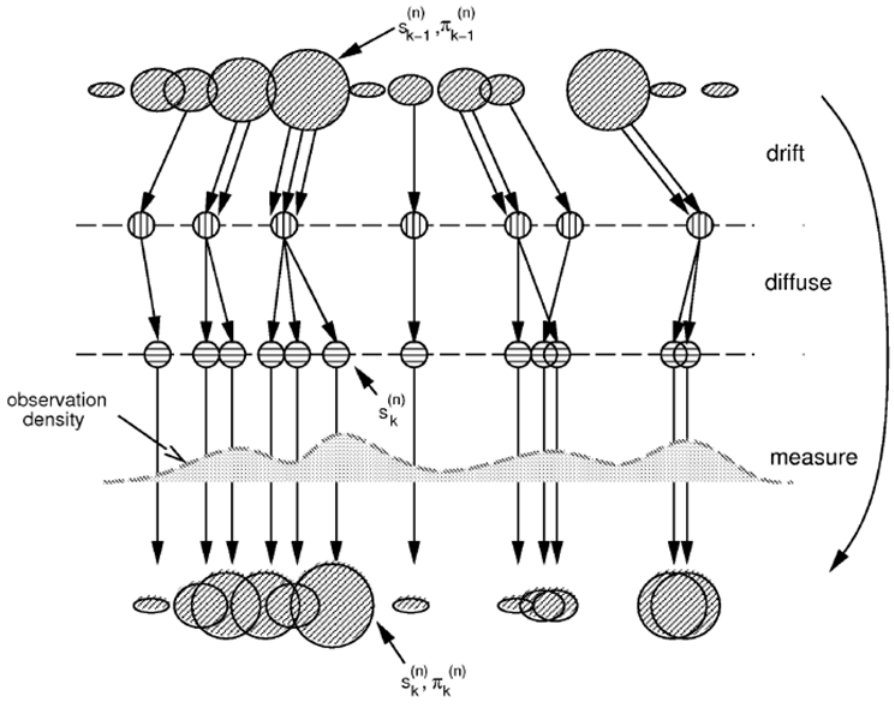

Figure 3.2: Illustration of one iteration of the Condensation algorithm [51]. Blob size represents the weight of a given sample (here it is denoted as $\pi_k^{(n)}$), while blob position represents the modelled state.

cluster transitions [50], which allows samples from the sample set to probabilistically switch shape clusters in every frame, based on a transition probability distribution, which we learn off-line for the purpose of modelling typical shape transitions (e.g., from left pose to frontal pose), in addition to the deterministic cluster switching that may happen during the image search algorithm (Alg. 2). This allows the samples to better explore the shape space, resulting in more accurate predictions.

In order to do this, we build a Markov transition matrix, $\mathbf{T}$, which we train on a sample video sequence containing many pose changes. This sequence is tracked by the face tracker and the shape cluster is recorded in every frame. For every pair of frames, we look at the shape cluster transitions and use them to populate the matrix, $\mathbf{T}$, so that the matrix element $\mathbf{T}(i, j)$ is incremented by 1, every time we encounter a transition from shape cluster i to shape cluster j. Each row of the transition matrix is

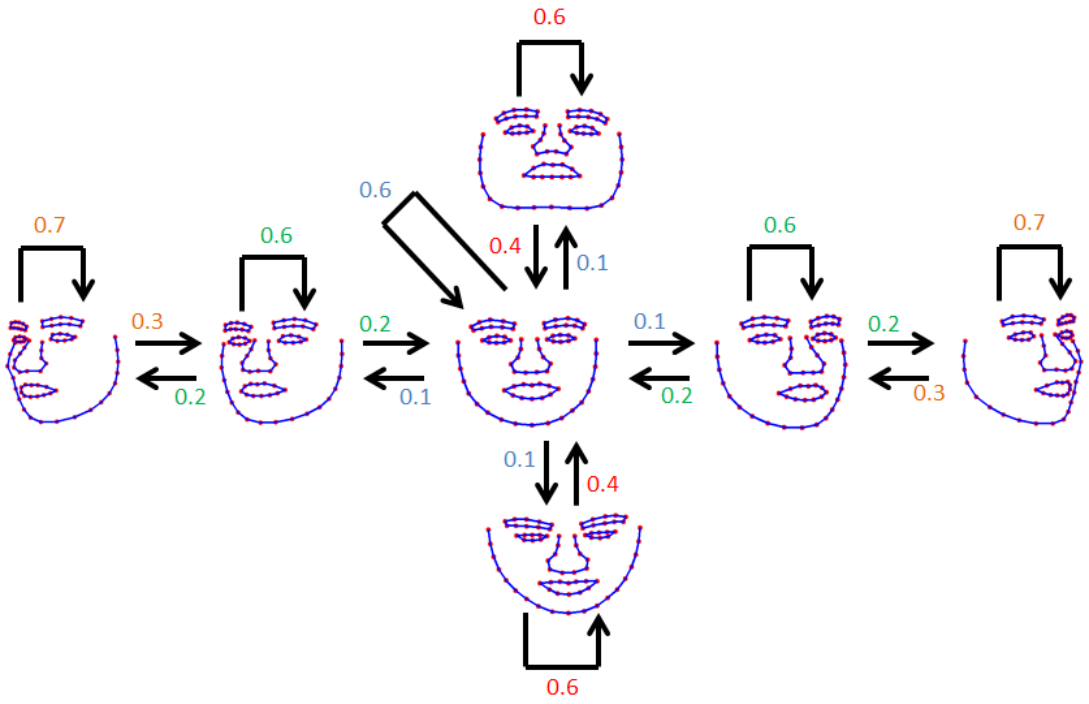

Figure 3.3: Illustration of the cluster transition probabilities used in our mixed state particle filter extension. Note that our method can be extended to include additional clusters for finer approximation of the non-linear shape manifold.

then normalized to sum to 1, using:

$$\mathbf{T}'(i,j) = \mathbf{T}(i,j) / \sum_{k} \mathbf{T}(i,k) \ , \tag{3.5}$$

so that the matrix element $\mathbf{T}'(i,j)$ gives the probability with which a shape from cluster i will transition to a shape in cluster j at the next time frame. For more efficient probability sampling, we also compute the cumulative transition probability matrix, using:

$$\mathbf{C}(i,j) = \sum_{k=1}^{j} \mathbf{T}'(i,j) \ . \tag{3.6}$$

The learned transition probabilities for the 7 clusters of our face tracking model are illustrated in Fig. 3.3.

3.4 Extended Tracking Algorithm

In this section we describe our novel tracking algorithm, which is based on the ASM face tracker [57] and extended with a mixed-state condensation filter for probabilistic tracking [50, 51].

Algorithm 3 Extended tracking algorithm

1. Initialize by running the regular ASM search (Alg. 2) to get initial state.
2. Use initial state to generate initial sample set, $\mathbf{S}_t$, for $t = 1$ based on the initial probability distribution, setting samples to uniform weight to get $\mathbf{S}^t = \{(1/n, \mathbf{s}_t^1), (1/n, \mathbf{s}_t^2), \ldots, (1/n, \mathbf{s}_t^n)\}$.
3. Get next frame. For each sample, run the ASM search (Alg. 2) for a few iterations (e.g., $n = 4$) to improve the fitting and then compute its observation likelihood using $p(\mathbf{z}_t|\mathbf{x}_t)$, updating sample weights accordingly.
4. Predict system state by first finding the mode cluster, and then calculating the weighted average of the samples in the mode cluster.
5. Re-sample from the sample set using the samples' weights, so that samples with high weight are more likely to be re-sampled. Set weights of re-sampled set to $1/n$.
6. Propagate re-sampled sample set in time by sampling from the transition probability matrix, $\mathbf{T}$. If sample transitions to the same cluster, use the dynamical model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Otherwise, cluster switching occurs and sample's state is set to mean shape of destination cluster.
7. If there are no more frames in the video stream, exit. Else, check if cumulative density is above threshold then go to step 3, otherwise go to step 1.

In landmark-based face tracking the state of the target can be represented by the coordinates of each tracked landmark. However, for complex PDMs having many landmarks, this can lead to a high-dimensional state very quickly. For face tracking we use 79 landmarks, so this would mean a 158-dimensional state vector. By exploiting the fact that for ASM face tracking the tracked landmarks lie on a shape manifold of reduced dimensionality, we can represent the system state using the shape cluster id (c), the global transformation parameters, i.e., translation (T_x, T_y), scaling (s), and orientation (θ), and the local shape deformation parameters, i.e., the eigen-encoding of the shape given the current cluster ($\mathbf{b} = [b_1, b_2, \ldots, b_q]^T$). This leads to a mixed-state for our

system, $\mathbf{x_t}$, at time t, given by $\mathbf{x_t} = [c, T_x, T_y, s, \theta, b_1, b_2, \ldots, b_q]^T$; obviously the length of this state will depend on the number of local deformation parameters used by each local cluster, which in our trained ASM model ranges from 35 to 57 eigen-parameters. We use a first order auto-regressive model as the dynamical model that will drive the global deformation parameters, while for the local deformation parameters we use a Gaussian noise model with zero mean and variance σ. Note that depending on the application and on the prior knowledge that we may have as to the dynamics of the modelled target, this can be changed in order to get better accuracy. However, we found that this choice of hybrid-dynamical model works well in practice for a range of videos we have tested it on.

The tracking algorithm extended with the modified condensation filter proceeds as follows. In the first frame we run Alg. 2 to get the starting state of the system. We use this initial state (i.e., global transformation parameters, local deformation parameters and cluster id) to generate an initial sample set with uniform weights, by sampling around the initial mixed-state based on a specified initial probability distribution, which may be application dependent. For the work presented here we used a uniform distribution centered around the initial state.

When we get the next frame, we run ASM search (Alg. 2) for $n = 4$ steps to improve the fit of each sample. Then we evaluate the observation likelihoods of each sample, $p(\mathbf{z}_t|\mathbf{x}_t)$, updating their weights (the observation model is explained in the next section). In order to avoid interpolating between peaks, we use the weights of each sample to find the cluster which has the highest total weight and then take the weighted average state of all samples within that cluster as the tracker's predicted state for this frame.

Next, the particles are re-sampled based on their weights, so that particles with higher weight are more likely to be selected in the re-sampling. The re-sampled particles are propagated forward in time one step, as follows. For each particle, sample from the cluster transition matrix to determine which cluster it should transition to. For example, generate a random number, p, uniformly over the range [0, 1], choosing the smallest j, such that $\mathbf{C}(i, j) > p$. This number j is the cluster to which the sample should transition to. If the current cluster of the sample is i, and if $i \neq j$, then cluster

switching occurs, in which case the sample's local deformation parameters are set to the average shape of the destination cluster, i.e. set $\mathbf{b} = \mathbf{0}^T$, while its global transformation parameters are set to the average value found in the training set plus a translation to center the sample around the last frame's prediction. If the current cluster of the sample is i, and if $i = j$, then no cluster switching occurs, and the sample's state is propagated one step in time using the dynamical model, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, which in our case is auto-regressive for the global parameters and $\mathcal{N}(0, \sigma)$ for the local deformation parameters.

When all samples are propagated, the cycle repeats for the next frame, until we get to the end of the video stream. In each step, we also check if the cumulative density of the sample set drops below a threshold, which could indicate a tracking failure. When this happens, we simply re-initialize the tracker and continue. The complete algorithm is summarized in Alg. 3.

3.5 Observation Model

Here we present our hierarchical observation model which is used to estimate the observation likelihood given the state of each particle in our sample set, $p(\mathbf{z}_t|\mathbf{x}_t)$. This likelihood will determine the weight, w_t^i, at time t that will be associated with the i^{th} sample, $\mathbf{s}_i^t$.

The observation likelihood function has the following three terms: (1) a term that rewards with a high/low weight those samples whose state represents landmarks with surrounding texture that matches well/poorly the local texture profiles learned during training of the ASM and are also computed for each iteration of the ASM search algorithm (see Alg. 2), (2) a term that rewards with a high/low weight those samples whose state represents facial component configurations (i.e., nose-tip, mouth, eyes, inner eyebrows) with surrounding texture that matches well/poorly the texture templates learned at the initialization of the tracker, and (3) a term that rewards with high/low weight those samples whose state represents a shape that is close to/far from the mean shape of the current local cluster, i.e., implausible shapes are penalized. This leads to

the following observation likelihood function:

$$p(\mathbf{z}_t|\mathbf{x}_t) \propto \quad (\alpha \times e^{-f(\mathbf{x}_t)/2\sigma_l^2}) \tag{3.7}$$

$$+ \quad (\beta \times e^{-(\sum_{i=1}^{6} g_i(\mathbf{x}_t)/2\sigma_t^2)}) \tag{3.8}$$

$$+ \quad (\gamma \times e^{-\mathbf{b}^T \Lambda^{-1} \mathbf{b}}) \ , \tag{3.9}$$

where Eq. 3.7 represents the local landmark likelihood, Eq. 3.7 represents the likelihood based on the six facial component templates and Eq. 3.9 represents the shape deformation likelihood. The scalars α, β and γ control how the components of the likelihood function are combined. For our experiments we used equal weights for α and γ and double weight for the template likelihood factor, i.e., $\alpha = \gamma = 1$ and $\beta = 2$. Similarly, the scalars σ_l and σ_t control the spread of the sub-likelihood functions. We used $\sigma_l = 1$ and $\sigma_t = 0.05$. The diagonal matrix, Λ, contains the eigenvalues for the current shape cluster learned during ASM training.

At initialization of the face tracker, we use the initial points to form a region of interest (ROI) around each of the following six facial components: nose-tip, mouth, left eye, right eye, left inner eyebrow, right inner eyebrow (see Fig. 3.4). These ROIs are used to extract a template, in the form of Edge Orientation Histograms (EOH), from each of these locations. EOH are simple to calculate and have been shown to be discriminative descriptors of region appearance for a variety of recognition tasks, such as hand gesture recognition [40]. EOH are calculated as follows. Fist we extract edges from the ROI using horizontal and vertical Sobel filters, K_x and K_y, which we convolve with the input image, I, to yield the gradient images, G_x and G_y:

$$G_x(x,y) = K_x * I(x,y) \ , G_y(x,y) = K_y * I(x,y) \ . \tag{3.10}$$

The gradient images are then used to compute the magnitude, $M(x,y)$, and orientation, θ, of the edges using:

$$M(x,y) = \sqrt{G_x^2(x,y) + G_y^2(x,y)} \ , \tag{3.11}$$

$$\theta = \arctan(G_y(x,y)/G_x(x,y)) \ . \tag{3.12}$$

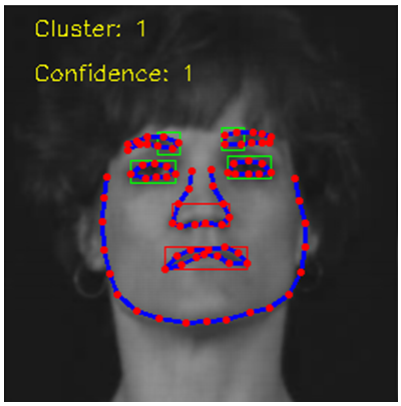

Figure 3.4: Illustration of the facial regions, which are used to calculate templates in the form of Edge Orientation Histograms.

We threshold the result so that edges with magnitude less than 5 are suppressed to 0. The histogram descriptor is formed by counting the edges into $k = 12$ bins with a vote equal to their magnitude. In order to encode the spatial distribution of edges within each template, instead of this simple histogram, we build a spatial pyramid [64] with $L = 2$ levels. The template descriptors can be updated dynamically so as to better represent the current target appearance, in case the target's appearance is changing quickly because of illumination changes or changes in facial expression. Template likelihoods, i.e., functions g_i, are computed using the spatial pyramid match kernel with histogram intersection (see Sec. 4.2.2 for more details).

3.6 Experimental Results

We have implemented the presented particle filter extension to the face tracking system of Kanaujia et al. [57]. Our goal was to create an improved tracking system, which can handle out-of-plane rotations, run reasonably quick, and have better handling of occlusions, especially those found in our ASL dataset, so that we can apply it to recognize dynamic events in ASL as well as dynamic events relevant to deception detection and in other domains.

Here we present experimental results, which validate our methods superiority over the original method, both qualitatively and quantitatively. For the reported experiments, we used 40 samples in the particle set, and we used $\alpha = \gamma = 1$ and $\beta = 2$

for the component weights of the observation model. We used 2 levels (i.e., $L = 1$) and 12 bins in the spatial pyramids of the edge orientation histogram (EOH) features and evaluated the template match likelihood using histogram intersection. The tracker of [57] achieved a running time of about 30 fps, while our extended face tracker run at about 5 fps (Intel Quad-Core 2.4GHz, 8GB RAM). Note, however, that in its implementation we have made no attempt to optimize it, except for optimization of the EOH feature calculation by means of integral images [93], since for the applications presented in this work all tracking and processing was done off-line and after all the video data has been collected. If desirable, the implementation can be sped up with GPU programming or multi-threading to parallelize the computations, since samples can be processed independent of each other and then merged again to get the weighted sample set.

Given that the most challenging video sequences occurred in our ASL dataset, we have selected from this dataset a relatively long sequence of 142 frames of resolution 640×480, where there is a long period of severe occlusion of the signer's eyebrows. In particular, in this sequence, the signer is performing a non-manual marker for a yes-no question, which involves raised eyebrows and the head jutted forward. The non-manual marker begins at frame 37 and ends at frame 105. Because of the signer's hairstyle, as soon as she begins performing the non-manual marker, her raised eyebrows get occluded behind her hair for more than 2 seconds until almost the very end of the video sequence.

In Fig. 3.5 we show a qualitative comparison of the two methods; our extended face tracker corresponds to the result of the top two rows. Initially both tracking methods perform well and track the facial landmarks and the facial expression of raised eyebrows. This is shown by an increasing curve in the graphs on the right (rows 1 and 3). The blue colors of the graph marks the time period during which there is no non-manual marker, while the red color of the graph marks the time period during which the non-manual marker of the yes-no question occurs. The superiority of our method is evident once the occlusion occurs at around frame 35. Rows 2 and 4 show that while our method (row 2) continues to track and correctly measure the eyebrow height of the raised eyebrows throughout the video sequence, thereby extracting correct

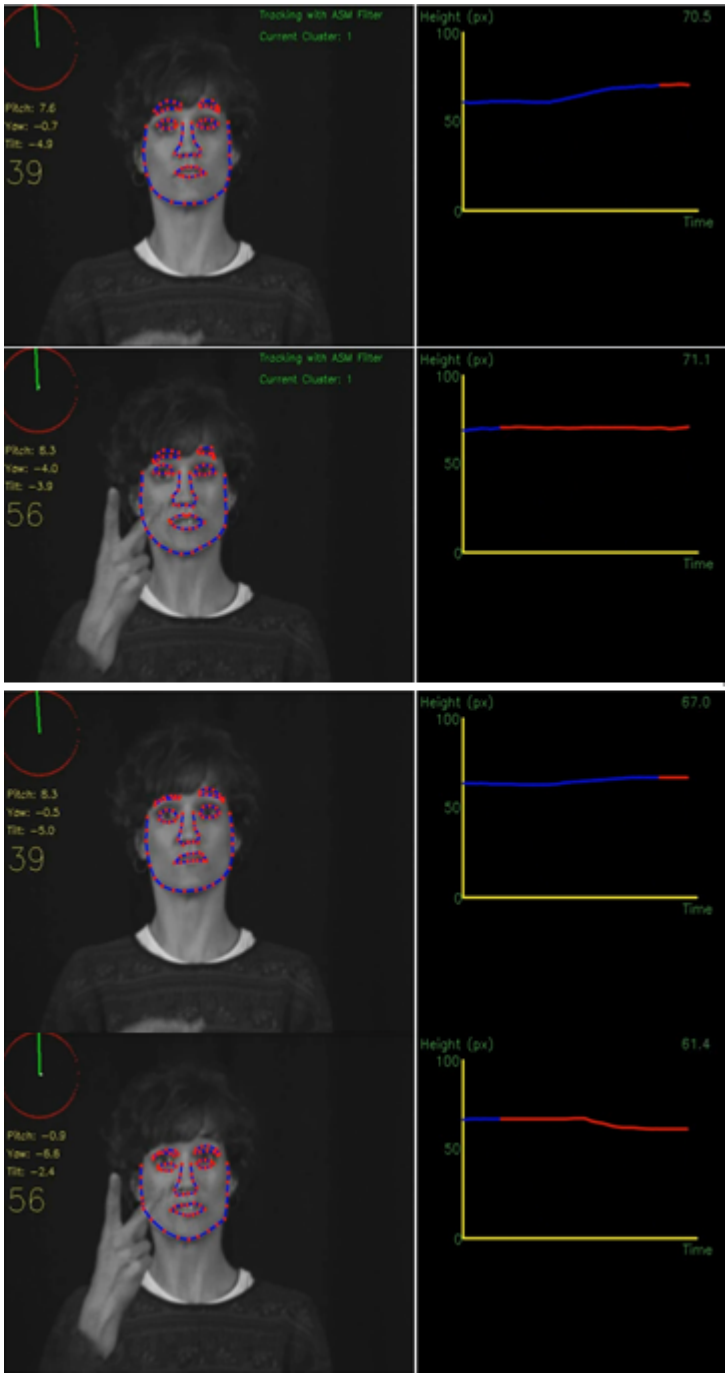

Figure 3.5: Comparison of our presented particle filter extension (top two rows) to Kanaujia et al.'s [57] face tracker (bottom two rows) under eyebrow occlusion. Note that our method stays locked on target, while Kanaujia et al.'s tracker drifts downwards shortly after the eyebrow occlusion occurs, and registers a sharp decrease in the graph of eyebrow height. Correct estimation of eyebrow is crucial for correct recognition of the non-manual component of ASL (see Sec. 4), which validates our choice of methods for the proposed extension.

features which can be used for accurate recognition the ASL non-manual markers, while Kanaujia et al.'s method [57] (row 4) fails to maintain track past the first few frames. In fact, once the eyebrows get completely occluded that face tracker drifts downwards resulting in inaccurate track of both the eyebrows and the eyes. What's worse, this drift is registered as a decrease in eyebrow height (instead of the expected increase which characterizes this type of non-manual marker), which means that use of this method without our extension for recognition of ASL non-manual markers would fail, since it would not be able to correctly capture the discriminative features of this recognition problem.

In addition to the qualitative comparison, we did a quantitative comparison where we manually annotated the positions of 9 key-points in each frame of the given video sequence. More specifically, these 9 key-points are: inner and outer corner points of each eye, inner and outer corner points of each eyebrow, and the nose-tip. We then calculated the root mean squared error (RMSE) for each method. The RMSE for both methods averaged over all 9 key-points is shown in Fig. 3.6. The red line is for the ASM tracker [57] and the blue dotted line is for our ASM tracker extended with the particle filter we described earlier. Note the jump in RMSE values for the tracker of [57] at around frame 35, which is where the occlusion begins. These high values of RMSE continue until about frame 135 when the eyebrows begin to become visible again. On the other hand, the RMSE of our method is much lower throughout the occlusion.

Similarly, we show additional plots with RMSE averaged over both eyes (Fig. 3.7), both eyebrows (Fig. 3.8) and the nose-tip (Fig. 3.9), where the same trend holds, except for the case of the nose-tip which both trackers track relatively well. Note that although the occlusion concerns the eyebrows, the RMSE plot for the eyes obtained from the result of [57] shows a similar pattern, because when the drifting happens during the occlusion, the prediction of the eyebrows drops to above the eyes, which in turn pushes the prediction of the eye below the actual eye position. These results are summarized in Table 3.1, where we clearly see the benefit of using our extended face tracker.

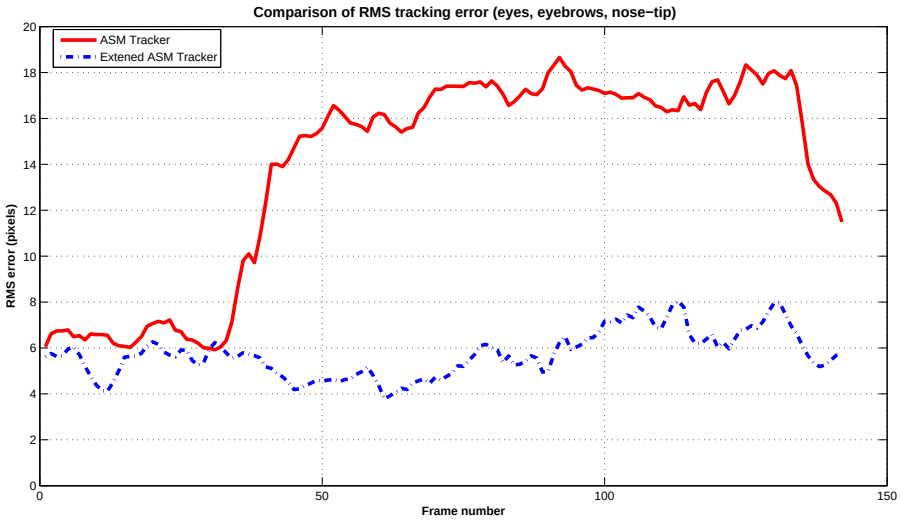

Figure 3.6: Plots of Root Mean Squared tracking error using the ASM face tracker with and without our particle filter extension. RMS error is averaged over 9 key-points (the two corners of each eye, the inner and outer left and right eyebrows and the nose-tip).

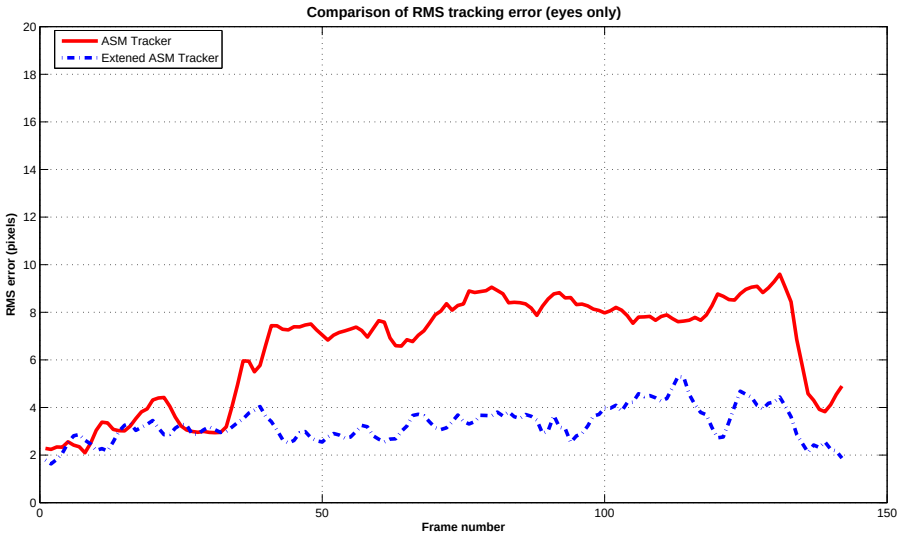

Figure 3.7: Plots of Root Mean Squared tracking error using the ASM face tracker with and without our particle filter extension. RMS error is averaged over 4 key-points (i.e., left and right corners of each eye).

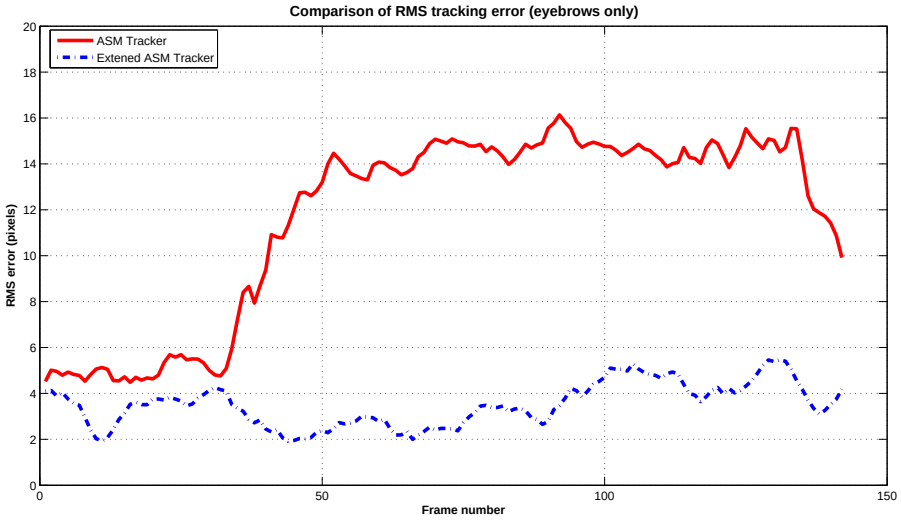

Figure 3.8: Plots of Root Mean Squared tracking error using the ASM face tracker with and without our particle filter extension. RMS error is averaged over 4 key-points (i.e., left and right corners of each eyebrow).

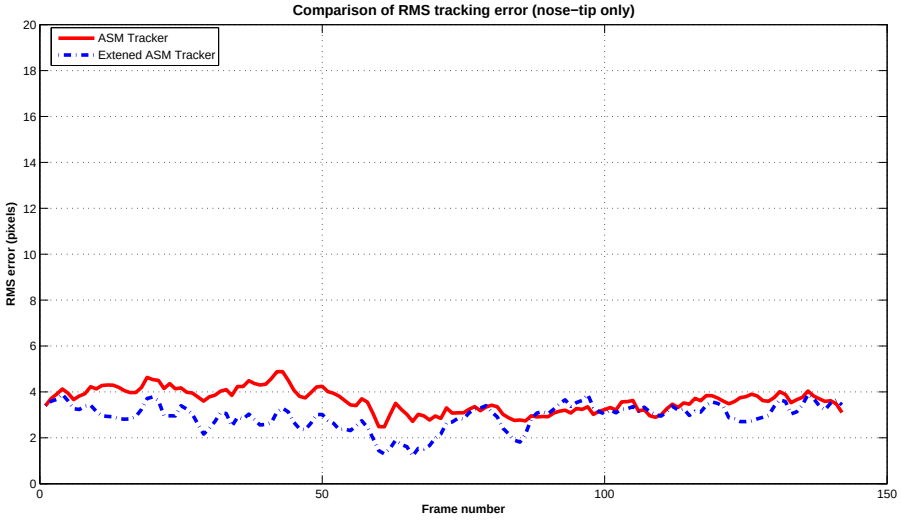

Figure 3.9: Plots of Root Mean Squared tracking error of the nose-tip using the ASM face tracker with and without our particle filter extension.

	Neutral	Onset	Eyebrows Raised	Offset	Overall
All Key-points (no PF)	6.5	7.2	16.6	14.2	13.8
All Key-points (with PF)	5.4	5.7	5.8	5.8	5.7
Eyes (no PF)	3.0	3.9	8.0	5.1	6.5
Eyes (with PF)	2.6	3.2	3.5	2.4	3.3
Eyebrows (no PF)	4.8	6.0	14.1	12.7	11.7
Eyebrows (with PF)	3.2	3.7	3.5	3.9	3.5
Nose-tip (no PF)	4.1	4.1	3.4	3.6	3.6
Nose-tip (with PF)	3.3	2.8	2.8	3.5	2.9

Table 3.1: Comparison of RMS tracking error (in pixels) with and without our particle filter extension during the four stages of an ASL grammatical facial expression (yes-no question) involving raised eyebrows, followed by the overall error in the sequence. The raised eyebrows get severely occluded by the signer's hair, causing the face tracker to lose track and drift downwards, while the tracker with our Particle Filter extension maintains accurate track of all important key-points. The video resolution was 640×480 and the sequence contained 142 frames.

3.7 Summary

Face tracking has numerous applications in the field of Human Computer Interaction and behavior understanding in general. However, face tracking is a difficult problem because the tracker must generalize to new faces, adapt to changing illumination, keep up with fast motions and pose changes, and tolerate target occlusion. In this chapter, we have presented our face tracking system, which extends the work of Kanaujia et al. [57], with a particle filter for probabilistic tracking via a dynamical system, probabilistic cluster transitions, and an observation model of hierarchical appearance and anthropometric constraints. Empirical evidence on a challenging video sequence shows that our system can handle out-of-plane head rotations and other shape local deformation in a probabilistic manner. It additionally handles occlusion of facial components, which is an essential requirement for the applications we present in the next chapters, with much lower root mean squared tracking error (RMSE) than the method we extended, both during the occlusion period and for the entire sequence overall.

Chapter 4

Recognition of Non-Manual Markers in ASL Video

In this chapter, we describe the application of our face tracking system to the problem of recognition of non-manual markers in video sequences of American Sign Language. We begin with some theoretical background about the nature of non-manual markers in ASL. We then first present a framework we have used for isolated recognition of non-manual markers in segmented ASL video, whereby we assume that we know the start and end times of a non-manual marker within a video sequence, and then an extension to it where we address feature misalignment and model the temporal patterns between neighboring video frames. Next we present a framework for continuous recognition, where we assume no knowledge about start and end times of non-manual markers in the video sequences, and then extend this with an image warping method that achieves head pose normalization of the computed features using a 3D face model.

4.1 Background

In Sec. 1.1.2 we have mentioned that facial expressions and head gestures convey critical grammatical information in ASL sentences, which help disambiguate sentences that differ only in their non-manual component. This section aims at providing additional information about the different classes of non-manual markers that we attempt to recognize in this work and the facial expressions and head gestures associated with them.

More specifically, in wh-questions (which involve phrases such as who, what, when, where, why, and how), the grammatical marking consists of lowered eyebrows and squinted eyes that occur either over the entire wh-question or solely over a wh-phrase that has moved to a sentence-final position. The possibilities are illustrated in the example ASL sentences of Figure 4.1. In this figure, labelled lines indicate the signs

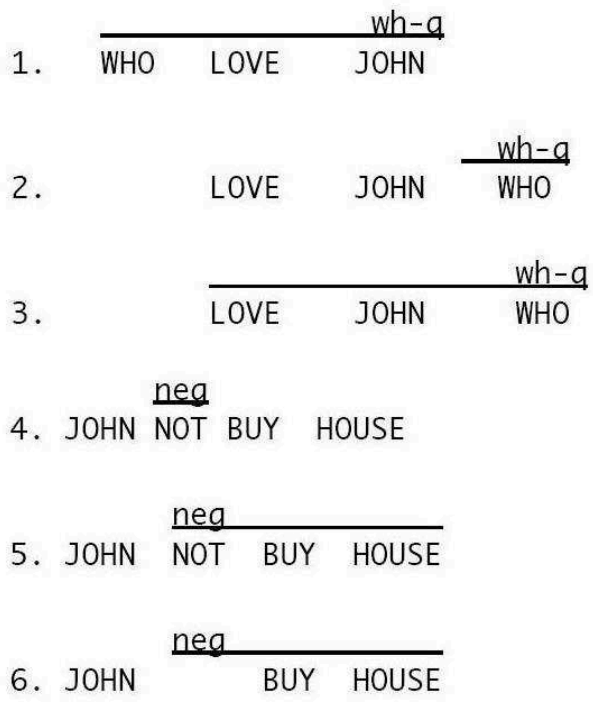

Figure 4.1: Several sample ASL sentences for negative and wh-question constructions, with English glosses representing the ASL signs.

with which the non-manual marking co-occurs. The first three examples would be translated in English as "Who loves John?". The intensity of this wh-question marking is greatest at the end of the sentence when it spreads over the entire question, as in Figure 4.1(3). In addition, there may be a slight, rapid side-to-side head shake over at least part of the domain of the wh-question marking.

With negation, there is a relatively slow side-to-side head shake that co-occurs with a manual sign of negation (such as NOT, NEVER), if there is one, and may extend over the scope of the negation, e.g., over the following verb phrase that is negated. These possibilities for translating an English sentence meaning "John did not buy a house" are illustrated in the bottom three examples of Figure 4.1(4-6). The intensity of this marking is greatest at the source of the syntactic feature being marked, as in wh-questions, but in a sentence like (5) or (6) of the same figure, this means that the intensity of the negative marking (including the amplitude of the head turns) is greatest at the left edge and diminishes as the marking continues. For further detail about distribution and intensity of non-manual grammatical markings, see [83].

Figure 4.2: Still samples of the ASL non-manual markers that we recognize with our methods: (first row) rhetorical questions; (second row) topics; (third row) conditional/when clauses; (fourth row) yes-no questions; (fifth row) negative statements; (sixth row) wh-questions.

Moreover, conditional/when sentences are two part constructions with the relevant non manual marking only over the first part (i.e., over the "conditional" or "when" clause). This is characterized by raised eyebrows, wide eyes, head forward (or back) and tilted to the side, followed by a pause, after which the eyebrows and head return to neutral position. Lastly, topics are characterized by raised eyebrows, wide eyes, head tilted back, and an optional nod, while rhetorical questions involve raised eyebrows, head tilted to side and usually tilted back but sometimes forward; followed by a pause and then an answer. Figure 4.2 shows sample still images of the classes of non-manual markers just discussed. In some cases the non-manual marker is easily recognizable from the still snapshot, e.g., wh-questions, while for most others (e.g., those involving raised eyebrows or a head shake) a sequence of images is required to recognize the marker.

4.2 Framework for Isolated Recognition

Using our extended face tracker (see Chap. 3), we accurately track the faces of American Sign Language (ASL) signers, localizing their facial components (e.g., eyes, eyebrows) and predicting their 3D head pose. Inspired by the work of Lazebnik on scene categorization [64], together with the popularity of "bag-of-words" models [65], we use spatial pyramids of features to detect lowered eyebrows and squinted eyes. We augment this information with the 3D head pose using Stacked Generalization and Majority Voting [108, 125], to recognize the presence of wh-question facial expressions in a video sequence. Additionally, we extend the idea of spatial pyramids to the temporal dimension, constructing pyramids of head pose derivatives (i.e., the change of head pose), for the recognition of head shakes that are characteristic of negative expressions. The detailed algorithm is described in Alg. 4. Next we describe the components used in this framework, followed by our experimental results [75, 84].

Algorithm 4 Algorithm for Isolated Recognition [75, 84]

1. Face tracking and pose estimation of video segment containing non-manual marker.
   (a) Feed video sequence into face tracker to localize and track signer's face.
   (b) Face tracker outputs (x,y) positions of 79 facial landmarks and the 3D head pose for each frame.
2. Feature Extraction for each tracked frame utilizing ASM tracker's output.
   (a) Compute bounding box of eyes and eyebrows and extract dense SIFT feature descriptors from it [69].
   (b) Soft quantize the SIFT descriptors and the head pose using separate feature codebooks [42].
   (c) Build pyramid representation of frames and video sequences.
      i. Build spatial pyramids of computed SIFT descriptors for each frame.
      ii. Build temporal pyramid of head pose derivatives for the entire sequence.
3. Recognize video sequences containing negative expressions using the temporal pyramid representation of pose derivatives and a Support Vector Machine (SVM) [13] with pyramid matching kernel [64].
4. Recognize video sequences containing wh-question expressions.
   (a) Use a stacked Support Vector Machine [108, 125], which combines the score obtained from classifying the spatial pyramid representation of SIFT descriptors and the score obtained from classifying the pose angle, to classify each frame in the video sequence.
   (b) Apply majority voting [108] on the results of the previous step, to classify the entire sequence based on the classification of each frame within the sequence (if the majority of the frames are classified as depicting a wh-question expression, the entire sequence is also classified as such).

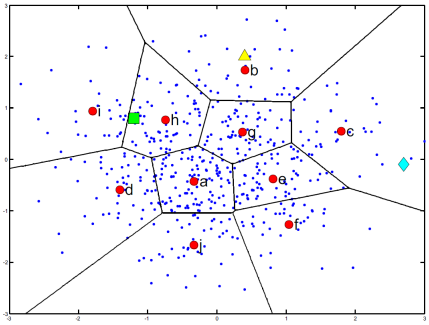

Figure 4.3: Hard quantization works well for points like the yellow triangle. However, the encoding of the green square is ambiguous, while that of the cyan diamond is implausible.

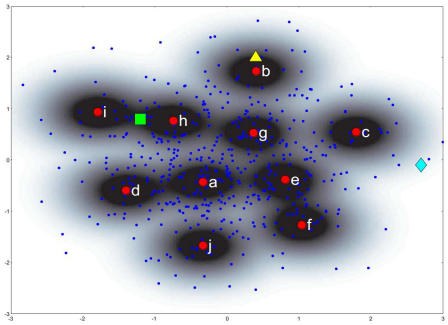

Figure 4.4: Soft quantization reduces the problem of codeword ambiguity by allowing prototypes to have a cloud of influence, instead of encoding features by a single prototype.

4.2.1 Codebook Construction

The codebook approach, inspired by the word-document representation used in text retrieval, was first applied to images in the work of Leung and Malik [65]. This approach allows classification of images by representing them as a bag of features, for example SIFT features [69], which are in turn represented as discrete prototypes [42]. Typically researchers use unsupervised clustering to obtain a codebook, $\mathbf{V}$, of prototypes, $\mathbf{v}_w$, from a random subset of the training data and label each feature by the index, w, of its best representing prototype, which minimizes some distance function e.g. Euclidean distance. Then they count how many times each prototype, $\mathbf{v}_w$ occurs in an image and stack these frequencies in a vector, which becomes the new compact representation of the image. This codebook encoding is essentially a histogram of the distribution of codebook prototypes within a given image and can later be used for classification purposes.

However, quantizing features in this manner creates problems. For example, if some feature is too distant from all available prototypes, either because the feature in question is an outlier or because there are not enough prototypes to adequately cover the feature space, forcing such a hard assignment could mean that the resulting encoding is implausible. Moreover, if a feature is very close to more than one prototype, it becomes ambiguous as to which one would represent it the best. Figure 4.3 illustrates

these issues. The obvious solution of simply increasing the codebook size bears with it an undesirable increase in both dimensionality and computational complexity. Instead, the authors of [42] overcome these problems of codeword plausibility and codeword ambiguity by employing ideas from kernel density estimation and allowing the prototypes to have a cloud of influence over the feature space. The shape of the influence cloud is controlled by the choice of kernel function and its scale parameter (see Figure 4.4 for an illustration). In particular, they propose a soft assignment of image features to prototypes, resulting in the following Kernel Codebook (KCB) encoding for each prototype, $\mathbf{v}_w$:

$$KCB(w) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(D(\mathbf{v}_w, \mathbf{x}_i)) \ , \ 1 \leq w \leq W, \tag{4.1}$$

where $KCB(w)$ is the value of the w^{th} bin of the histogram encoding, W is the codebook size, N is the number of features in the image, $\mathbf{x}_i$ is the i^{th} feature, $D(\mathbf{v}_w, \mathbf{x}_i))$ is the distance of the w^{th} prototype from the i^{th} feature, and σ is the smoothing parameter of kernel K. In this way, multiple prototypes can contribute to the encoding of each feature, with their contribution weighed in inverse proportion to their kernel distance from it. In our work, we adopt this method of soft quantization, setting K to be a Gaussian kernel with standard deviation σ, and using Euclidean distance as our distance metric, $D(\mathbf{v}_w, \mathbf{x}_i)$.

4.2.2 Pyramid Representation

After we extract and softly quantize [42] the discriminative SIFT features of each frame, we utilize the work on pyramid representation of Lazebnik et al. [64], which enables us to model the spatial relationships among features and also provides the means for measuring feature similarity between frames, using a pyramid match kernel.

Denote the set of quantized features extracted from two frames as X and Y. To build a pyramid with L levels, for each level $l = 0, 1, ..., L$, we divide the frame into an imaginary grid of $2^{2\times l}$ cells, along both the x and y dimensions, so that the cells in level l are bigger than the cells in level $l + 1$ above it. We histogram the quantized

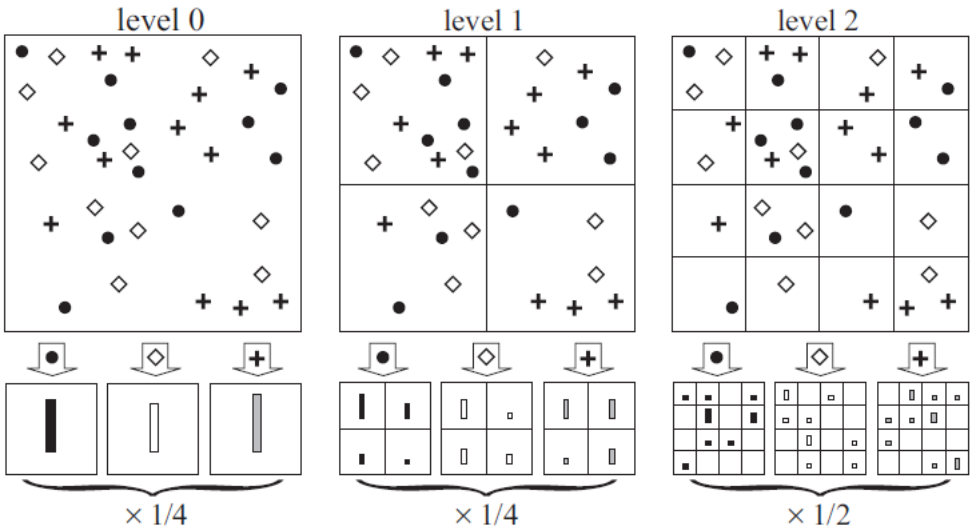

Figure 4.5: Toy illustration of spatial pyramid construction [64], where, for simplicity, we assume there are only 3 codewords (circle, diamond, cross). The top part shows the successive subdivisions of the image into different resolution levels. For each level and for each spatial bin in that level, we count the frequency of each codeword, forming histograms weighed according to equation (4.3). These weights are shown in the bottom of the figure.

features that fall in each cell (for each SIFT feature descriptor we know its position within the frame it came from), yielding separate histograms for each cell in each of the L levels. These histograms represent the feature distribution of a particular cell, in terms of the relative frequency of occurrence of each feature prototype within that cell. Because cells at different levels have different sizes, their histograms are computed over image subregions of different sizes, yielding an image representation of different levels of resolution. The topmost layer, having the smallest sized cells, forms the most detailed representation of the feature distribution within an eye region, while the bottommost layer the least detailed. Collectively, the histograms at each level form the pyramid representation of the feature distribution within an image, which is effectively a concatenated vector of the bin values of all the histograms in the pyramid (see Fig. 4.5 for an illustration).

Figure 4.6 shows two spatial pyramid representations extracted from video sequences containing different facial expressions. The pyramid on the left corresponds to a frame in which the signer was producing a wh-question expression, while the pyramid on

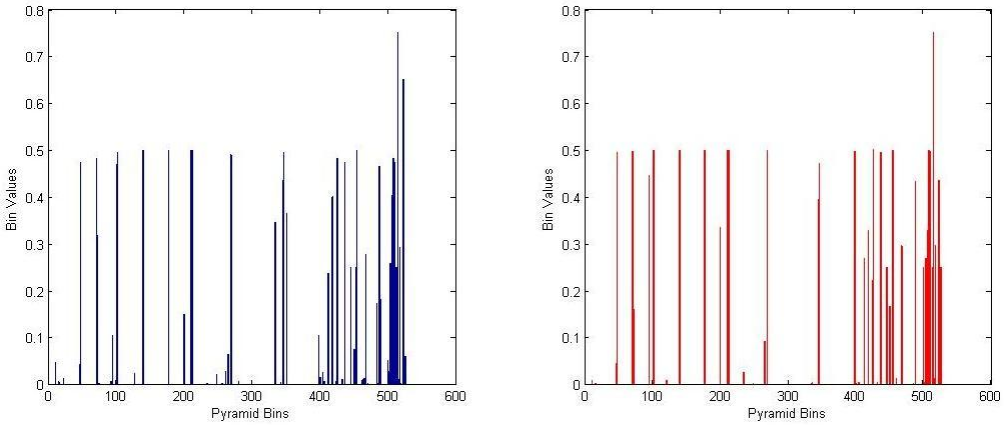

Figure 4.6: Spatial pyramids of SIFT descriptors (50-word codebook, $\sigma = 0.2$). Pyramid levels are decreasing with increasing bin index. Left plot is for a wh-question. Right plot is for a negative expression.

the right comes from a video of a negative expression. Examining the two plots, the difference in the pyramids is evident, especially in the levels of finer resolution (finer resolution bins are on the left). The input frames, together with the tracked faces and the extracted eye regions, that generated these spatial pyramids are shown in Figure 4.7.

In order to measure the distance between the feature sets X and Y, and eventually measure the dissimilarity in appearance between any pair of frames,we just need to compare their pyramid representations, essentially meaning comparing the bins of these histograms to see how much they match. Similar to [64], we measure histogram similarity at each level l, using the histogram intersection function presented in the work of Swain and Ballard [104] and defined as:

$$I(H_X^l, H_Y^l) = \sum_{j=1}^{C} \min\left(H_X^l(j), H_Y^l(j)\right) \ , \tag{4.2}$$

where H_X^l and H_Y^l are the histogram representations of the two frames at level l, C is the number of cells at level l, while $H_X^l(j)$ and $H_Y^l(j)$ are the respective histograms of frames X and Y in the j^{th} cell of level l.

Since higher levels are of a finer resolution, it is intuitive to weigh the similarity match of cells in these levels with a higher weight than that used for the lower levels

Figure 4.7: First column shows the input frame, second column shows the tracked face with the estimated 3D pose, and third column shows the extracted eye and eyebrow region. The top signer is producing a wh-question, while the bottom signer is producing a negative expression.

of coarser resolution. Moreover, if a match is found at a level l, it will also be found in the coarser level $l-1$, so when comparing feature sets, we should only consider the new matches found at each level. This leads to the following match kernel for spatial pyramids having L levels:

$$K^L(X,Y) = \frac{1}{2^L} I^0 + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}} I^l \ , \tag{4.3}$$

where I^0 is the intersection score at level 0 and I^l is the intersection score at level l [64].

Furthermore, we propose a natural extension of this pyramid representation to the temporal domain. The ASM face tracker predicts the head pose in each frame. We compute the change in yaw angle between successive frames and softly quantize the yaw derivatives using a codebook that we compute from a random subset of the training set. Then we construct a temporal pyramid for each video, by dividing a sequence of frames into cells, in a similar fashion as done for spatial pyramids and using the same match kernel. In this way, we form a representation which allows us to detect the head

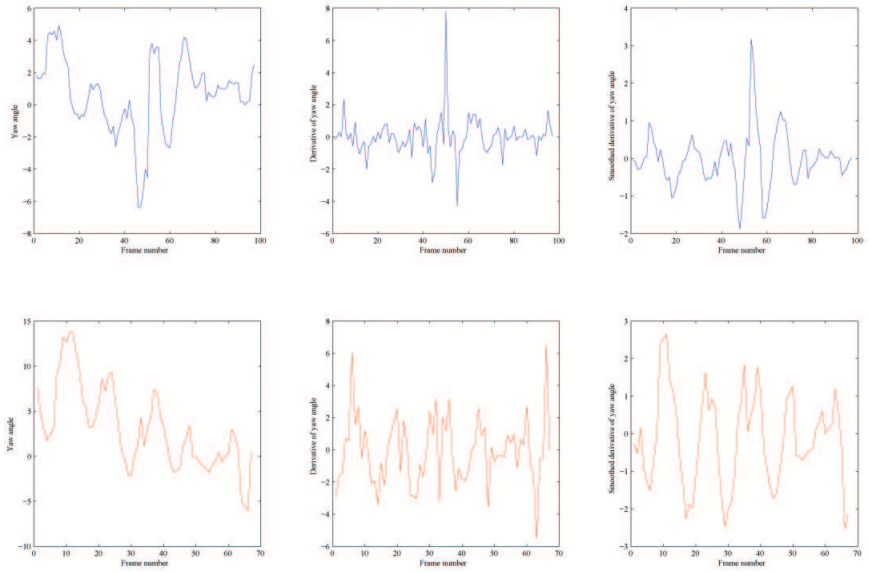

Figure 4.8: Sample plots of yaw angles, yaw derivatives and smoothed derivatives for two video sequences of different class. Top row plots are from a wh-question. Bottom plots are from a negative construction.

shake of a signer. This is because we expect to see a distinct uniform pattern of yaw angle derivatives resulting from a head shake during a negative expression, which is distinct from the pattern of yaw derivatives resulting from other ASL expressions. This difference in yaw angle derivative patterns is illustrated in Figure 4.8.

4.2.3 Overview of Support Vector Machines

A Support Vector Machine (SVM) is a popular classifier with excellent generalization properties [13, 113]. The key idea is to learn a decision boundary or margin, in the form of a hyperplane that passes through the training data, so that the data is correctly classified, and that the distance of every training point to this hyperplane is maximized. Hence, SVMs are often called margin maximizing classifiers. In the case of non-separable classes, the SVM learns an optimal hyperplane which is again at a maximum distance from as many training data points as possible, while at the same minimizing the misclassification rate. SVMs can also learn non-linear hyperplanes by utilizing a kernel function to map the data points to a higher dimension; this is known as the "kernel trick" [13, 113].

Let $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$ be the training data, such that $\mathbf{x_i} \in \Re^m$, and let $\mathbf{Y} = \{y_1, y_2, ..., y_n\}$ be the corresponding training class labels, such that $y_i \in \{-1, +1\}$. Also assume that we are first dealing with separable classes, so that there exists an optimal hyperplane which correctly separates the data points into the two classes and at the same time maximizes the margin, i.e. the distance of any data point to the decision boundary. Let $\mathbf{w}$ be the vector normal to this hyperplane and threshold b be its hyper-intercept. Then the training data is correctly satisfied if the following constraints are satisfied:

$$\mathbf{w^T x_i} + b \geq +1 \quad \text{for } y_i = +1 \tag{4.4}$$

$$\mathbf{w^T x_i} + b \leq -1 \quad \text{for } y_i = -1 \ , \tag{4.5}$$

which can be combined into one set of inequalities [13]:

$$y_i(\mathbf{w^T x_i} + b) - 1 \geq 0 \ , \ \ i = 1, ..., N \ . \tag{4.6}$$

Since the data is assumed to be separable, there exist an infinite number of hyperplanes that satisfy equation (4.6), so in order to find the margin maximizing hyperplane we must also add the constraint that $||\mathbf{w}||^2$ is minimized [13]. The combined constraints can be formulated as the following constrained minimization problem:

$$\min \{\frac{1}{2}||\mathbf{w}||^2\} \ , \text{ subject to } y_i(\mathbf{w^T x_i} + b) - 1 \geq 0 \ , \ \ i = 1, ..., N \ . \tag{4.7}$$

Skipping the details, the solution to this non-linear (quadratic) optimization problem, subject to a set of linear inequalities, can be found by finding the solution to the Karush–Kuhn–Tucker (KKT) conditions of its Lagrangian, using the primal–dual path following method [13].

In the case that we are dealing with non-separable classes, the formulation is similar, except for the introduction of positive slack variables ξ_i into the constraints of equations (4.4) and (4.5):

$$\mathbf{w^T x_i} + b \geq +1 - \xi_i \quad \text{for } y_i = +1 \tag{4.8}$$

$$\mathbf{w^T x_i} + b \leq -1 + \xi_i \quad \text{for } y_i = -1 \ , \tag{4.9}$$

$$\xi_i \geq 0, \ \ \forall i \ , \tag{4.10}$$

which lead to the following updated constrained optimization problem (where C is a complexity parameter) [13], solved in a similar way:

$$\min \{\frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i\} \ , \text{ subject to } y_i(\mathbf{w^T x_i} + b) - 1 \geq \xi_i \text{ and } \xi_i \geq 0 \ , \ \ i = 1, ..., N \ . \tag{4.11}$$

Once the optimization problem is solved, we can compute the signed distance (margin) of any test point $\mathbf{x}$ to the decision boundary. The sign of this margin tells us on which side of the margin the test point lies, hence we can assign it a class label $f(\mathbf{x}) = \{-1, +1\}$ using:

$$f(\mathbf{x}) = \begin{cases} 1 & \mathbf{w}^T\mathbf{x} + b > 0 \\ -1 & \mathbf{w}^T\mathbf{x} + b < 0 \end{cases} \ , \tag{4.12}$$

while points that lie on the decision boundary (i.e. $\mathbf{w}^T\mathbf{x} + b = 0$) can be classified as either positive or negative instances, depending on the bias chosen for a particular application. In the case that a kernel function is used to train a non-linear SVM then the classification rules for a test point $\mathbf{x}$ become:

$$f(\mathbf{x}) = \begin{cases} 1 & (\sum_i^{N_s} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x})) + b > 0 \\ -1 & (\sum_i^{N_s} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x})) + b < 0 \end{cases} \ , \tag{4.13}$$

where N_s is the number of support vectors $\mathbf{x_i}$ (see [13] for a more detailed derivation and examples). In this work, we adopt the non-linear formulation of SVMs for non-separable classes, where the non-linear kernel is the pyramid match kernel described in Section 4.2.2.

4.2.4 Experimental Results

The Boston University American Sign Language Linguistic Research Project (ASLLRP) dataset used for the research reported here consists of 15 spontaneous short narratives

	Training	Testing	Total
Wh-question	25	11	36
Non-Wh-question	25	11	36
Total	50	22	72
Negative Expression	22	10	32
Non-Negative Expression	22	10	32
Total	44	20	64

Table 4.1: Dataset Composition.

plus over 400 additional elicited utterances collected from several native signers of ASL [84]. Synchronized video cameras captured the signing from multiple viewpoints (two stereoscopic front views plus a side view and a close-up of the face). The data were annotated using SignStream®, software (http://www.bu.edu/asllrp/SignStream/), developed specifically for linguistic annotation of visual language data [82]. The annotations include identification of start and end frames of individual signs as well as labelling of facial expressions and head movements that have grammatical significance.

In our experiments we used the close up view of the face only from isolated utterances. We selected a total of 36 video sequences showing wh-questions and 32 sequences showing negative expressions. These formed our set of positive examples for each of the two classes. An equal number of negative examples were collected by randomly selecting video sequences from different classes (e.g. conditional–when, topic–focus, yes–no questions, etc.). We then randomly split our two datasets of wh-questions and negative expressions into a training and validation set, and into a test set, ensuring that both sets contained data from different signers. The duration of the video sequences ranged from about 3 seconds to 12 seconds. The shortest duration of a wh-expression was about 0.8 seconds (23 frames) and the longest was about 5.1 seconds (153 frames). The shortest duration of a negative expression was about 0.6 seconds (19 frames) and the longest was about 4.4 seconds (131 frames). The training sets contained about 70% of the total data, while the remaining data formed the testing set. Table 4.1 shows the dataset composition in more detail.

We used our extended face tracker (see Chap. 3) to track the signer's face in each sequence and extract their eye region, as well as predict their 3D head pose. Figure

	Precision	Recall	Accuracy
Stacked Wh-question	91.7%	100%	95.5%
SIFT Wh-question	90.9%	90.9%	90.9%
Pose Wh-question	63.6%	63.6%	63.6%
Negative Expression	90.9%	100.0%	95.0%

Table 4.2: Performance metrics for isolated recognition.

	Predicted as Negative	Predicted as Non-Negative
True Negative	10	0
True Non-Negative	1	9

Table 4.3: Confusion matrix for isolated recognition of negative expressions.

4.7 shows sample results of tracking, pose prediction and localization of the eye region. The pose angle predictions were smoothed with a one-sided Gaussian filter with $\sigma = 2$ and a length of 7 frames, so that the pose in a given frame was a weighted combination of the pose predictions in that frame and of those in the 6 frames before it, in order to filter out noise. Pose angle derivatives were computed, as the difference in pose angle between two successive frames, and then a random subset was used to construct a codebook of 75 codewords using soft assignment [42] with a Gaussian kernel and $\sigma = 0.1$ (larger size codebooks did not achieve better recognition). Temporal pyramids with three levels (i.e. $L = 2$) were then constructed for each video sequence and a Support Vector Machine (SVM) with the pyramid match kernel discussed in Sec. 4.2.2 was trained and used to classify the test set sequences into negative and non-negative expressions. The complexity parameter of the SVM model, C, was chosen with 5-fold cross validation to avoid over–fitting the training set.

The SVM classifier achieved a precision accuracy of 90.9% and a recall rate of 100%, with an overall recognition accuracy of 95%. Using more levels in the temporal pyramid hurt the performance. Here, by recognition accuracy we refer to the percentage of instances in the training set that were correctly classified (i.e. $\frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{N}}$). Precision is the ratio $\frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}}$ and recall is the ratio $\frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}}$, where N stands for the total number of test set instances, tp stands for true positive, fp for false positive and fn stands for false negative. The detailed classification results are shown in Table 4.2, while Table

	$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$
25 keywords	0.830	0.798	0.845	0.858	0.892	0.751
50 keywords	0.840	0.837	0.855	0.882	0.874	0.701
100 keywords	0.850	0.854	0.889	0.861	0.841	0.703
200 keywords	0.867	0.864	0.890	0.879	0.803	0.696

Table 4.4: Area under the ROC curve (AUC) of the SVM models used to recognize frames containing wh-expressions, trained only on the spatial pyramid features, obtained using different combinations of dictionary sizes and kernel scale, σ.

4.3 shows the confusion matrix.

Similarly, from the localized eye regions we have extracted dense SIFT features [69], which we also quantized using soft assignment [42] with a Gaussian kernel. Sample spatial pyramids with three levels (i.e. $L = 2$) extracted from frames in which different signers are producing different grammatical constructs, are shown in Figure 4.6. As done for negative expression recognition, we trained an SVM model with a pyramid match kernel and used cross validation to choose the value of parameter C. In this case, however, we first used the SVM to classify each frame within a sequence and then used Majority Voting [108] to decide the label of each sequence based on the majority label assigned to its constituent frames. We experimented with different codebook sizes and different kernel scales, σ, in order to obtain the best combination and we found that a codebook of 100 words and $\sigma = 0.2$ performed the best.

Figure 4.9 shows the corresponding ROC curves for the classifier corresponding to each combination of codebook size and kernel scale, while Table 4.4 shows the area under the corresponding ROC curve. As expected we observe that as the codebook size increases, so does the recognition accuracy, because we have an increasing number of prototypes providing an improved cover of the feature space. Increasing the kernel scale compensates for the inadequate cover of the feature space by a small number of prototypes but once the scale gets too large for a particular codebook size, the influence clouds become increasingly larger, allowing features to accept representation influence by distant prototypes resulting in an incorrect encoding. Table 4.5 summarizes the classification accuracy on the test sequences using majority voting on the class labels of their constituent frames as predicted by the SVM models.

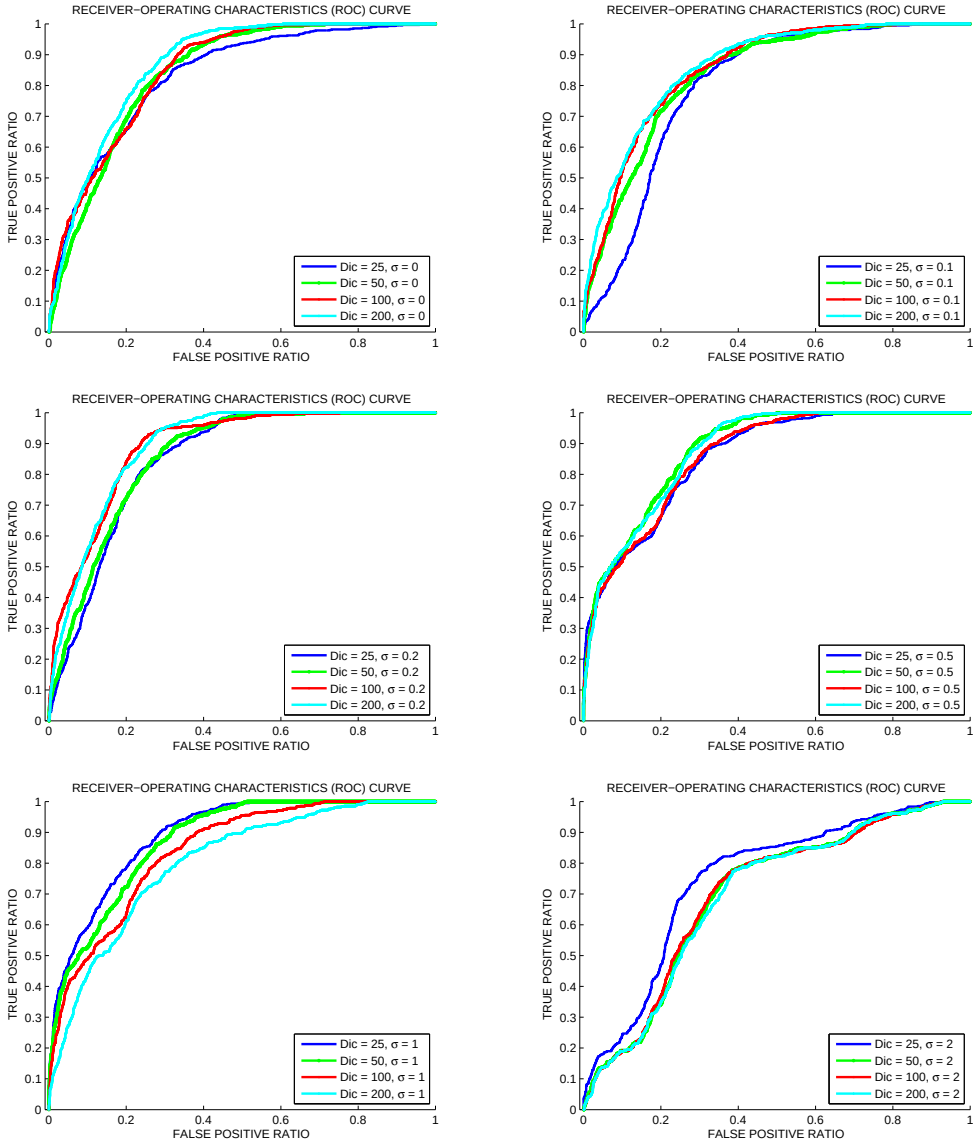

Figure 4.9: ROC curves of wh-expression recognition (statically on a frame level) using an SVM trained on spatial pyramid features only, under various combinations of dictionary size and kernel scale, σ.

	$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$
25 keywords	90.9%	90.9%	95.5%	81.8%	81.8%	77.3%
50 keywords	95.5%	90.9%	95.5%	81.8%	81.8%	81.8%
100 keywords	95.5%	90.9%	100%	81.8%	81.8%	81.8%
200 keywords	90.9%	90.9%	100%	81.8%	77.3%	81.8%

Table 4.5: Majority Voting recognition accuracy of wh-expressions in isolated sequences, using an SVM trained only on spatial pyramid features, obtained by different combinations of dictionary sizes and kernel scale, σ.

	$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$
25 keywords	90.9%	95.5%	90.9%	86.4%	86.4%	81.8%
50 keywords	90.9%	86.4%	100%	90.9%	90.9%	81.8%
100 keywords	90.9%	86.4%	95.5%	90.9%	86.4%	81.8%
200 keywords	90.9%	81.8%	95.5%	86.4%	81.8%	81.8%

Table 4.6: Majority Voting recognition accuracy of wh-expressions for isolated sequences, using a stacked SVM combining spatial pyramid features, obtained using different combinations of dictionary sizes and kernel scale, σ, and head pose features.

In our previous work, [75, 84], we found that the head pose of the signer might be correlated with the grammatical facial expression being made. In particular, we observed that the signer tilted their head backwards when signing a wh-question. Using this finding, we implemented two separate base SVM classifiers to classify individual test frames into wh-questions and non-wh-questions; one used only eye appearance features (i.e. SIFT pyramids), which we called SIFT–SVM model, and one used only head pose features, namely zero, first, second and third order pitch angle differences, which we called Pose–SVM model. The un-thresholded predictions of the two SVMs for each frame were then combined in a Stacked SVM framework, [108, 125], by a trained meta–classifier which output the final label for each frame. Stacking [108, 125] allows us to learn to smartly combine the individual predictions of multiple base classifiers in order to improve classification accuracy, by utilizing the specific expert knowledge learned during training by each of the base classifiers. The meta–classifier was also an SVM which used a radial basis function (RBF) kernel; we refer to this model as the Stacked–SVM. The complexity parameter, C, and the kernel scale, σ, of the Pose–SVM and of the Stacked–SVM were determined by cross validation. For isolated recognition, the predictions of the individual frames within each segmented sequence were aggregated using majority voting, in order to obtain the final prediction for the sequence. Table 4.6 summarizes the recognition accuracy of the Stacked–SVM models on the test sequences. We notice that there is a general improvement in recognition accuracy when combining eye region appearance and head pose features of the signer, which in turn allows us to learn smaller size dictionaries (compare Tables 4.5 and 4.6). Table 4.2 summarizes the performance metrics for isolated recognition of wh-questions.

4.3 Modelling Temporal Dependencies and Misalignment

Although the method of the previous section achieved good recognition performance on two classes of non-manual markers, for differentiation of non-manual markings that differ very subtly from one another, it is going to be crucial to combine multiple evidence (e.g., use head positions and movements, appearance features around the nose, etc.). Therefore, in this section we present an extension to our framework for isolated recognition, which allows us to recognize non-manual markings associated with wh-questions and negative sentences, as well as conditional/when clauses, yes/no questions and topics. The last three classes all involve raised eyebrows, so to be able to distinguish them, it will be necessary to employ additional features and model the temporal feature transitions between neighboring frames.

More specifically, the proposed extension differs from the method in the previous section in the following ways. First, once we track the facial landmarks, we focus on an extended rectangular region of interest (ROI), which includes the eyes, eyebrows and nose, so as to capture a wider range of upper face expressions, e.g., nose wrinkling and cheek tensing. Second, we divide this ROI into a set of smaller patches (henceforth referred to as parts), which correspond roughly to areas of the face relevant for these specific grammatical expressions, e.g., inner and outer eyebrows. We extract from each part a histogram of Local Binary Patterns (LBP) [89]. These are effective for texture classification [99, 132], faster to compute and more robust to illumination variations than SIFT, which is used in [75]. Third, we handle feature misalignment, arising from tracking inaccuracies and partial facial occlusions, by computing a Multiple Instance Feature (MIF) [68] for each part. Fourth, in addition to the head pose and texture features per frame, we explicitly calculate eyebrow height. The final feature descriptor is augmented with a "summary" of the features of future and past frames sampled at regular intervals in the neighbourhood of the current frame, which we call "Oracle Features" (see Figure 4.10). This representation aims to encode the dynamic nature of facial expressions and head gestures encountered in non-manual grammatical markers. Lastly, by utilizing a discriminative, margin-maximizing, Hidden Markov Support

Vector Machine (HMSVM) [1, 53] our method outperforms generative Hidden Markov Models (HMMs) [94], which can over-fit the training data in the absence of sufficient training examples. Each new component of the proposed extension is discussed in the following subsections, followed by experimental results that validate our methodology.

4.3.1 Tracking eyebrow height and head pose

From the tracked landmarks we can compute the 2D position of a signer's left, (x_L, y_L), and right inner eyebrows, (x_R, y_R), and their nose-tip, (x_N, y_N), in each frame. For inner eyebrow calculation we use the 4 innermost eyebrow landmarks, while for the nose-tip we use the lower 8 nose landmarks. The eyebrow height at time t, denoted as h^t, is derived as the average Euclidean distance between the nose-tip and each inner eyebrow:

$$h^t = \frac{1}{2} \times \left(\sqrt{(x_L - x_N)^2 + (y_L - y_N)^2} + \sqrt{(x_R - x_N)^2 + (y_R - y_N)^2} \right) \ . \tag{4.14}$$

For robustness to tracking noise, we filter the computed (x, y) positions of the key points (eyebrows and nose-tip) using a Kalman filter [56], assuming linear state dynamics with Gaussian noise, $\mathbf{w}$. The system state, $\mathbf{x}_t$, includes the position, $\mathbf{p}_t = [x_L, y_L, x_R, y_R, x_N, y_N]^T$, and the velocity, $\mathbf{v}_t = [\dot{x}_L, \dot{y}_L, \dot{x}_R, \dot{y}_R, \dot{x}_N, \dot{y}_N]^T$, of these key points at time t. The dynamic process is governed by:

$$\mathbf{x}_{t+1} = A_k \mathbf{x}_t + \mathbf{w}_t \ , \tag{4.15}$$

with

$$A_{t+1} = \begin{pmatrix} 1 & \delta t \\ 0 & 1 \end{pmatrix} \text{ and } \mathbf{w}_t \sim \mathcal{N}(0, Q) \ . \tag{4.16}$$

The observation process is modelled as:

$$\mathbf{z}_t = H \mathbf{x}_t + \mathbf{u}_t \ , \tag{4.17}$$

where $H = [1, 0]$, $\mathbf{z}_t$ is the observation as obtained by the face tracker and $\mathbf{u}_t \sim \mathcal{N}(0, R)$

is the observation noise at time t. A similar model is also used to filter the predicted head pose. In this case the state vector includes the 3D head pose, $\mathbf{a}_t = [a_P, a_Y, a_T]^T$, and the head pose velocity $\dot{\mathbf{a}}_t = [\dot{a}_P, \dot{a}_Y, \dot{a}_T]^T$, where P, Y and T stand for pitch, yaw and tilt angles respectively.

4.3.2 Texture Features

Once we track the signer's head, we compute a bounding box of the tracked landmarks around the eyes, eyebrows and nose, forming an extended ROI from which we compute Local Binary Patterns (LBP) [89]. Put simply, LBPs are binary codes that characterize the texture in the neighbourhood of a pixel by thresholding the value of each neighbour by the gray-scale value of the central pixel (set to 1 if larger, set to 0 otherwise) and interpreting the pattern as a binary number, which is converted to a decimal code. Typically, LBP codes are first computed for each pixel in an image patch and then the normalized histogram of LBP codes is generated and used as a texture descriptor of the patch.

4.3.3 Oracle Features

Facial expressions and gestures are dynamic processes, especially those that have a grammatical meaning in ASL. It is often difficult even for ASL signers to detect non-manual markers using static frames alone. For example, one key component of the non-manual marking of negation is a head shake, whose presence in a sequence cannot be detected solely by looking at the head pose in any single frame. Instead, one needs to have available a "snapshot" of the variation of head yaw angle over time, in order to detect the turning of the head in one way and then in the opposite way.

Therefore, in order to strengthen the representational power of all features (texture MIF, head pose, eyebrow height), we encode information from neighbouring frames. For each frame we sample the feature values at regular offsets (sample points) from the current frame (anchor point). Before sampling, we compute a weighted average (by means of a Gaussian curve) of the feature value in a small window around the anchor and each sample point. This is illustrated in Figure 4.10 where an example anchor

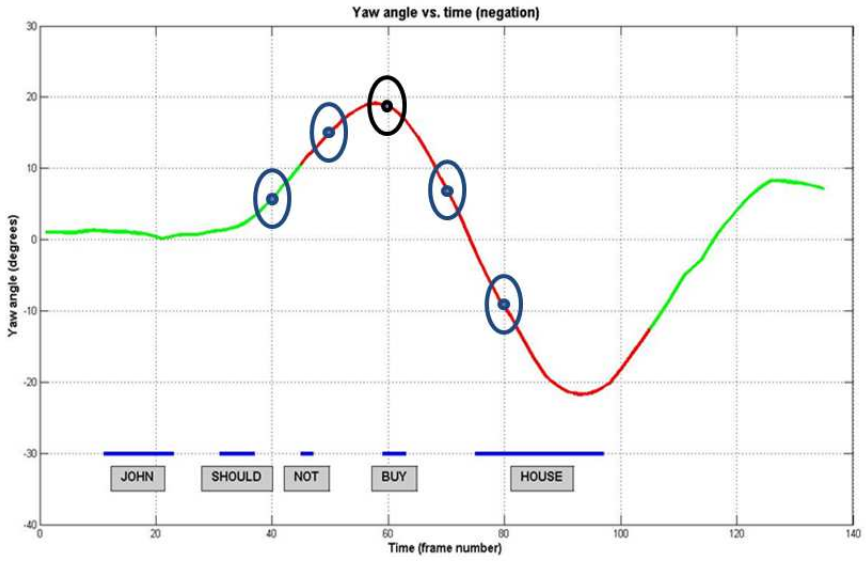

Figure 4.10: Plot of head yaw angle over time for a sequence containing negation (red segment marks when the non-manual marker occurs), also illustrating computation of yaw oracle features for frame 60 (see Section 4.3.3).

point is shown in black and example sample points are shown in blue. The ellipses indicate the size of the averaging neighbourhoods. Thus, the final feature descriptor of each frame is formed by combining features in that frame with the features obtained from the neighbourhood of the respective sample points. We refer to these augmented feature vectors as "Oracle features" because for every frame they encode the dynamic evolution of feature values. We show that this richer feature representation allows our method to achieve higher classification accuracy (see Section 4.3.6).

4.3.4 Overview of Multiple Instance Features

Feature misalignment sometimes occurs; i.e., the same features do not always fire up in all positive detection windows, often because of object pose variation. Lin *et al.* [68] introduced Multiple Instance Features (MIF) for boosted learning of part-based human detectors, where an initial boosting seeds the location of an object part from translated candidates, and then multiple instance boosting pursues an aggregated feature for each part. So an MIF is an aggregation function of instances. More specifically, given a

classifier, C, it is the aggregated output, y, of a function, f, of classification scores, $\{y_j\}_{j=1}^{J}$, on multiple instances, $\{x_j\}_{j=1}^{J}$:

$$y = f(y_1, y_2, ..., y_J) = f(C(x_1), C(x_2), ..., C(x_J)). \tag{4.18}$$

Each bag, x_i, consists of a set of instances, $\{x_{ij}\}_{j=1}^{N}$. For each classifier C, the score y_{ij} of an instance x_{ij} can be computed as: $y_{ij} = C(x_{ij})$. The probability of an instance x_{ij} to be positive is given by the logistic function: $p_{ij} = \frac{1}{1+e^{-y_{ij}}}$. In [71], the multiple instance learning problem is formulated as the maximization of diverse density, which measures the intersection of the positive bags minus the union of the negative bags.

The diverse density is probabilistically modelled using a Noisy-OR model to harness the multiple instance learning problem. The probability that a bag x_i is positive is formulated as $p_i = 1 - \prod_{j=1}^{N_i}(1 - p_{ij})$. The Noisy-OR model means the probability of the bag to be positive is high when this bag includes at least one instance with high probability to be positive, otherwise the bag is negative when all the instances inside have low probability of being positive. Following [68], the geometric mean is applied to avoid the numerical issues when N_i is large, so the formula is modified to $p_i = 1 - \prod_{j=1}^{N_i}(1 - p_{ij})^{1/N_i}$. The multiple instance aggregated score y_i is computed from the instance scores y_i as:

$$y_i = \log((\prod_{j=1}^{N_i}(1 + e^{y_{ij}})^{1/N_i}) - 1), \tag{4.19}$$

which comes from the logistic relation between p_i and y_i: $p_i = \frac{1}{1+e^{-y_i}}$. In this paper each y_i is an MIF of texture, obtained by learning weak classifiers (decision tree stumps) on the LBP histogram bins of a part (a part is a patch within the face ROI). See [68] for further details.

4.3.5 Overview of Hidden Markov Support Vector Machine

In the traditional supervised classification setting, we have a set of labelled training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \Re^d$ is the d-dimensional feature vector of training sample

i and $y_i \in \Re$ is its corresponding class label. The goal is to learn a mapping function from inputs to outputs $F\colon \Re^d \rightarrow \Re$ that minimizes some loss function, typically a 0/1 loss. In the sequence tagging problem we have sequences of feature vectors and for each one we have a sequence of corresponding outputs: $D = \{(\mathbf{x}_i^j, y_i^j)|j = 1, \ldots, J\}_{i=1}^N$, where J is the length of the i^{th} sequence. Note that sequences need not have the same length. The goal in this setting is to predict the class labels of all instance within each sequence.

A popular model used in sequence tagging problems (most notably for speech recognition) is the Hidden Markov Model (HMM) [94]. Despite its success, the HMM has certain limitations. First of all, it assumes conditional independence between observations when given the current state; an assumption that can be too restrictive for certain problems where there are complex feature interactions. Secondly, HMMs are generative models. During their non-discriminative training, the goal is to learn model parameters that maximize the likelihood of fitting the given training data, instead of optimizing for accurate classification (although recently there has been interest in alternative methods for training [124]).

Altun *et al.* [1] proposed the Hidden Markov Support Vector Machine (HMSVM), which, like the HMM, models the interactions between features and class labels, as well as interaction between neighbouring labels within a sequence. Unlike HMMs, the HMSVM model is trained in a discriminative margin-maximizing learning procedure. This means that it can achieve better generalization performance on test data, hence higher accuracy. Similar to the standard Support Vector Machine (SVM) [13], the HMSVM can also learn non-linear discriminant functions via the kernel trick.

Given a feature sequence $\mathbf{x} = \{\mathbf{x}^j\}_{j=1}^J$, where $\mathbf{x}^j$ are instances within the sequence the model predicts the corresponding tag sequence $\mathbf{y} = \{y^j\}_{j=1}^J$ using [1]:

$$\mathbf{y} = \arg\max_{\mathbf{y}\in\mathcal{Y}} \left(\sum_{j=1}^{J} \left(\sum_{k=1}^{K} < \mathbf{x}_j, \mathbf{w}_{y_{j-k},\ldots,y_K} > + < \phi_{\text{trans}}(y_{j-k}, \ldots, y_K), \mathbf{w}_{\text{trans}} >\right)\right) , \tag{4.20}$$

where $\mathbf{w}_{y_{j-k},\ldots,y_K}$ is an emission weight vector modelling interactions between features

and k^{th}order observations, and $\mathbf{w}_{\text{trans}}$ is the transition weight vector modelling transitions between neighbouring tags. Discriminative training aims to minimize the number of misclassified tags, while maximizing the separation margin, hence the training objective is [1]:

$$\min\{\tfrac{1}{2}\mathbf{w}^T\mathbf{w} + \tfrac{c}{J}\textstyle\sum_{j=1}^{J}\xi_j\} \tag{4.21}$$

$$\text{s.t.:} \quad z_j(\mathbf{y})(<\mathbf{w}, \Phi(\mathbf{x}_j, \mathbf{y}) > +\theta_j) \geq 1 - \xi_j \xi_j \geq 0 \ , \forall j = 1, \ldots, J \ , \forall \mathbf{y} \in \mathcal{Y} \ , \tag{4.22}$$

where c is a parameter that controls the penalty of misclassification trading off training error and margin size. Joachims et al. [53] proposed the cutting-plane algorithm which offers a significant speed-up in the training time of HMSVMs over the original working set algorithm of [1]. In our framework, from each frame in each segmented sequence, we use the oracle feature representation of the eyebrow height, the head pose and the multiple instance texture features, with their corresponding class label, and train a one-vs-all HMSVM model. Sequences in our training set contained no overlapping non-manual markers, so we only needed one model to tag each frame in the segmented sequence. Despite this, our method can easily generalize to sequences with overlapping non-manual markers by training n one-vs-all models (one for each class) and running them in parallel on each sequence.

4.3.6 Experimental Results

From the corpus of the Boston University American Sign Language Linguistic Research Project (ASLLRP) dataset, we selected training and testing sets of 32 and 13 video clips, respectively, of isolated utterances, extracting the segments containing non-manual markers of the classes of interest. Certain sequences contained multiple non-manual markers but there was no overlap between them. The exact composition of these sets per class is shown in Table 4.7. Both sets contained three different native signers.

Using the methods described in previous sections [77], we tracked the signer's head,

Class	Training Set	Testing Set
C/W	4 (292)	2 (158)
Neg	8 (532)	4 (258)
Top	9 (249)	4 (86)
Wh	8 (492)	5 (351)
Y/N	4 (262)	2 (172)

Table 4.7: Number of segmented sequences per class in our dataset (total number of frames in parenthesis)

extracting their head pose and computing their eyebrow height. These were post-processed with a Kalman filter for more accurate tracking. From the filtered head pose, we compute the head pose derivative per frame, to avoid learning a dependence on the initial head position of a signer. Eyebrow height is also normalized by the height in the first frame of each sequence and then we compute the height derivative, in order to normalize for subjects of different face proportions and distance from the camera. For each frame we compute oracle features as explained in Section 4.3.3. We use 5 sample points, offset at 0, +5, +10, +15 and +20 pixels from the current frame respectively, averaged over a 5 frame window, resulting in a 20-dimensional descriptor of head pose (pitch, yaw, tilt) and height variation per frame.

Before extracting texture features from the face ROI, we align all images, rotating frames by the average of the tilt angle and the angle between the centroids of the two eyes, as computed from the ASM landmarks. Faces were normalized by cropping frames to 64×64 pixels [106]. The face ROI is divided into a 4x4 cell grid with each cell being 16×16 pixels. From each cell we compute normalized histograms of uniform LBP features [89] using 8 samples and a radius of 1 pixel. For purposes of computing MIF [68], we consider each cell being one facial part (so we have 16 parts per frame) and translate each cell in a regular grid around its original position, computing additional LBP features. The collection of features for a given part form a bag of instances which we convert to a 5-dimensional MIF score, one for each class of non-manual markers. The idea is that if a positive part, with respect to a class label, is misaligned (as a result of tracking error or partial occlusion), as we translate it around its neighbourhood and compute instances of LBP features, at least one of these instances will capture features

True Class	Predicted Class				
	C/W	Neg	Top	Wh-Q	Y/N
C/W	100%	0	0	0	0
Neg	0	75%	0	25%	0
Top	0	0	100%	0	0
Wh-Q	0	0	0	80%	20%
Y/N	0	0	0	0	100%

Table 4.8: Confusion matrix of HMSVM segmented recognition using oracle features of LBP-MIF, head pose and eyebrow height.

	% Correct classification
HMM	70.6%
HMSVM	88.2%
HMSVM + non-MIF LBP	82.4%
HMSVM + MIF LBP + non-oracle	76.5%

Table 4.9: Evaluation of models showing the benefit of discriminative HMSVM with the proposed feature representation that handles feature dynamics and feature misalignment.

from a correct part placement, and the bag will still be positive for that class. As in the case of head pose and eyebrow height, we compute oracle features for the LBP MIF. Here, to avoid increasing feature dimensionality too much, we only use 3 sample points, offset at 0, +5 and +10 pixels from the current frame respectively, also averaged over a 5 frame window, resulting in a 240-dimensional texture descriptor.

The three sets of features (pose, height and texture) are concatenated into one feature vector and we train an HMSVM. Because of our small training set, we first optimize the parameter c using 3-fold cross validation on the training set, ensuring that each fold contains at least one sequence from each class, before evaluating on the test set. The recognition accuracy of the HMSVM model is summarized in Table 4.8. Analysis of the results revealed that for the wh-question sequence that is misclassified as a yes/no question the signer's head is rotated to the side, causing an incorrect estimation of the eyebrow height. Most importantly, this rotation causes a significant change in the appearance of the face ROI since most of the training images are frontal views. We expect to be able to overcome this problem by using training data that includes such cases of non-frontal faces. Additionally, our method mistakes a negative sequence for

a wh-question. In this sequence there is a clear head shake that our framework can capture and which is characteristic of negation. However, there is a head-shake – albeit somewhat different in character – that frequently occurs with wh-questions, as well as some degree of furrowing of the brows that occurs with both constructions. The model failed on this case, possibly because of insufficient training examples exhibiting this combination of eyebrow appearance and head shaking.

In order to compare the HMSVM with the HMM, we also trained 5 HMMs, one for each class, classifying test sequences as belonging to the class whose HMM yields the highest probability. The number of states of each HMM was decided using 3-fold cross validation. Results are shown in Table 4.9. Note that with our small dataset, the generative HMM fails to outperform the discriminative HMSVM model. In the same table we also show the result of an experiment where we used oracle pose and height features with non-MIF oracle LBP features and an HMSVM recognizer (HMSVM + non-MIF LBP). Note that this model performs worse, showing that the MIF indeed help improve accuracy. Using non-oracle features (HMSVM + MIF LBP + non-oracle) also hurts performance, as expected, given the dynamic dependence of features relevant to recognition of facial expressions and non-manual ASL grammatical markers in particular.

Therefore, the presented framework is successful in isolated recognition of a wider range of classes of non-manual markers. As the our results show, the key to the success of our method lies both in the discriminative recognition model (HMSVM) as well as in the rich feature representation that encodes feature dynamics and is able to handle feature misalignment. In the following, sections we describe an analogous framework for continuous recognition.

4.4 Framework for Continuous Recognition

In this section we present our framework [76] for continuous recognition, where we assume no knowledge about start and end times of non-manual markers in the video sequence. This method extends prior work [75, 84], in which the signer's head is tracked

and appearance features, in the form of spatial pyramids [64] of SIFT descriptors [69], are extracted from the region of interest, in the following ways. First, we extract additional shape features in the form of spatial pyramids of histograms of oriented gradients (PHOG) [7]. Second, we use spectral clustering [85], measuring the affinity using the Spatial Pyramid Match Kernel (SPMK) in Equ. 4.3, introduced by [64]. This reduces the dimensions of the augmented appearance and shape feature vectors. Lastly, we use Hidden Markov Models (HMMs) [94] to learn the dynamic feature transitions that occur during production of each class of non-manual markers. A summary of the algorithm is given in Alg. 5. Next we describe the components of our framework followed by our experimental results.

Algorithm 5 General algorithm for Continuous Recognition [76]

1. Face tracking on continuous video stream
2. Feature Extraction
   (a) Compute appearance features from ROI (PHOG and PSIFT) [7, 64, 69].
   (b) Compute head pose features.
   (c) Estimate eyebrow height (distance between eyebrow and eyes).
3. Get embedding of each set of appearance features using Spectral Clustering [4, 85].
4. Use trained HMM models [94] with sliding window for continuous recognition of video stream.

4.4.1 Overview of Feature Extraction

Once a frame is tracked and we have obtained the 2D position of the facial landmarks, we extract dense SIFT descriptors over a regular grid from the eye region. We cluster the SIFT descriptors of a random subset of the training frames, to obtain a codebook of prototypes and then encode all other descriptors by the index of their nearest prototype. As in the method for isolated recognition, we encode the spatial distribution of these features within the region of interest using spatial pyramids (see Sec. 4.2.2), to yield the spatial pyramid SIFT representation or PSIFT for short. The dissimilarity in appearance between any pair of frames, is measured by comparing the bins of these

histograms to see how much they match, using the weighted Spatial Pyramid Match Kernel (SPMK) given by Equ. 4.3 [104, 47, 64].

In order to get a more discriminative representation of the region of interest, we choose to combine the above mentioned appearance descriptor with a shape descriptor. In particular, we choose to use the pyramidal Histogram of Oriented Gradients (PHOG) used by the authors of [7]. PHOG descriptors are obtained by first applying a Canny edge filter to suppress weak edges. Then we quantize the gradient orientations of pixels into uniform bins, with each pixel's vote being proportional to the magnitude of its gradient, followed by construction of a spatial pyramid of HOG descriptors to get the final descriptor for the region. We compute PHOG features in the same way, but for measuring PHOG similarity we use the weighted SPMK in Equ. 4.3.

4.4.2 Overview of Hidden Markov Models

Hidden Markov Models (HMM) are statistical tools for modelling time series data [94]. Having been successfully used in speech recognition applications, researchers have also utilized them for recognizing handwriting, gestures, facial expressions, and of course sign language [102, 112, 117], among other things. In the following section we provide a brief overview of the theory behind HMMs ([94] provides a more thorough treatment).

HMM Definition

An HMM consists of a set of N distinct states, $Q = \{S_1, S_2, ..., S_N\}$. At any given time the system is in one of these N states and each state S_i has an associated initial probability π_i, which represents the probability that the system starts in state S_i.

At regular time intervals, it makes a transition from its current state at time t, denoted as Q_t, to its next state, denoted as Q_{t+1}, both of which can take any value from the above set of states Q. These transitions are governed by the HMM's transition probabilities, which, in the case of homogeneous HMMs, are invariant over time. For example, the probability of transitioning from S_i at time t to S_j at time $t+1$, is denoted by a_{ij}, i.e. $P(Q_{t+1} = S_j | Q_t = S_i) = a_{ij}$, therefore $\sum_i P(Q_{t+1} = S_j | Q_t = S_i) = \sum_i a_{ij} = 1$. It should be emphasized that for first order HMMs, state transitions obey

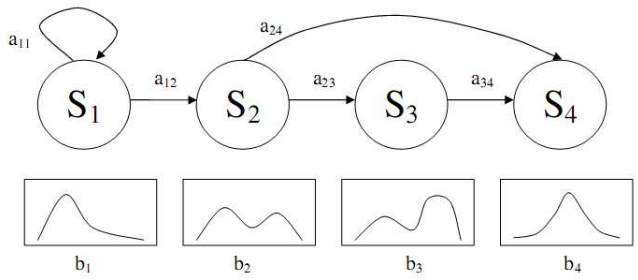

Figure 4.11: An example of a Left to right HMM with 4 states. Links represent the permissible state transitions, while link labels correspond to the transition probabilities. The plot under each state represents its emission distribution.

the Markov property, which means that the probability of the transition from Q_t to Q_{t+1} only depends on the value of Q_t, i.e. $P(Q_{t+1} = S_j|Q_t = S_i) = P(Q_{t+1} = S_j|Q_t = S_i, Q_{t-1}, Q_{t-2}..., Q_1) = a_{ij}$. These transition probabilities are collectively represented by an $N \times N$ matrix, A, where the entry in the i^{th} row and j^{th} column is a_{ij}. HMMs in which a transition to an already visited state is not allowed, i.e. $a_{ij} > 0$, implies $j \geq i$, are called left-to-right, while HMMs in which all possible transitions are allowed are called ergodic. Left to right HMMs are typically used in speech recognition to model phonemes. In this paper we also utilize this topology for our HMM models.

Additionally, at time t the current state $Q_t = S_i$ generates an observation $O_t = k \in \Omega$, which follows an observation probability distribution (also called emission probability distribution), denoted as $b_i(k)$, i.e. $P(O_t = k|Q_t = S_i) = b_i(k)$, therefore $\sum_k P(O_t = k|Q_t = S_i) = \sum_k b_i(k) = 1$. From this, another HMM model assumption should become clear: observations are only dependent on the current state that generated them.

Thus, an HMM model λ is parameterized as $\lambda = (\pi, A, B)$. Figure 4.11 shows a left-to-right HMM model with 4 states. The links between states illustrate the permissible state transitions, while the plots underneath each state depict the respective emission probability distributions. Such multi-modal distributions are typically modelled by a Gaussian mixture model [94].

The Basic HMM Problems

Having defined HMM models we now look at the three basic problems that HMM theory deals with [94]:

1. Given a sequence of observations $O = O_1, O_2, ..., O_T$, compute the probability that it was generated by a given HMM model λ, i.e. compute $P(O|\lambda)$.

2. Given a sequence of observations $O = O_1, O_2, ..., O_T$ and an HMM model λ, compute the most probable state sequence $Q = Q_1, Q_2, ..., Q_T$ that produced O, i.e. find Q that maximizes $P(Q, O|\lambda)$.

3. Given a sequence of observations $O = O_1, O_2, ..., O_T$ and an HMM model λ, adjust its model parameters, λ, so as to maximize $P(O|\lambda)$.

The first problem is the evaluation problem and is useful for the recognition of an unknown input sequence with a set of trained HMMs, where, in our case, each HMM corresponds to a particular class of grammatical facial expressions e.g. Wh-expressions. For each HMM, λ, we compute the likelihood of the given observation sequence, $P(O|\lambda)$. Then the input sequence is labelled with the class corresponding to the HMM that yielded the highest likelihood. Rabiner [94] proposed the recursive forward-backward algorithm which computes this observation likelihood, $P(O|\lambda)$, in $O(N^2T)$ time. The algorithm makes use of the two model assumptions mentioned in Section 4.4.2, namely the observation independence assumption and the Markov property of the state transitions. More specifically, let $Q = Q_1, Q_2, ..., Q_T$ be a state sequence of the HMM model λ, and define the forward variable $\alpha_t(i) = P(O_1, O_2, ..., O_T, Q_t = S_i|\lambda)$, for $1 \leq i \leq N$, where N is the number of model states as before. Then:

$$\alpha_1(i) = \pi_i b_i(O_1) \ , \tag{4.23}$$

$$\alpha_{t+1}(i) = b_i(O_{t+1}) \sum_{j=1}^{N} \alpha_t(j) a_{ji} \ \ 1 \leq t \leq T-1, \tag{4.24}$$

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \ . \tag{4.25}$$

Equation 4.23 is the initialization of the recursive procedure, Equ. 4.24 is the induction step and Equ. 4.25 is the termination step which computes the desired observation likelihood.

The solution to the second problem for a given observation sequence O and a model λ, can be thought of as "un-hiding" the hidden state sequence Q that generated O. It corresponds to finding the most probable state sequence Q which could have emitted O, i.e. finding Q which maximizes $P(O, Q|\lambda)$. Rabiner [94] suggests solving this problem using a dynamic programming technique, namely the Viterbi algorithm. Define the variable $\delta_t(i)$ to correspond to the maximum probability of all state paths that end up in state S_i at time t (i.e. $Q_t = S_i$):

$$\delta_t(i) = \max_{Q_1, Q_2, ... Q_{t-1}} P(Q_1, Q_2, ..., Q_t = S_i, O|\lambda) \ . \tag{4.26}$$

The Viterbi algorithm is then summarized by the following equations:

$$\delta_1(i) = \pi_i b_i(O_1) \ , \tag{4.27}$$

$$\delta_{t+1}(i) = b_i(O_{t+1}) \cdot \max_{1 \leq j \leq N} \{\delta_t(j) a_{ji}\} \ , \tag{4.28}$$

$$\max_Q P(Q, O|\lambda) = \max_{1 \leq i \leq N} \{\delta_T(i)\} \ . \tag{4.29}$$

Similar to the forward-backward algorithm, Equ. 4.27 initializes the procedure, the induction step is represented by Equ. 4.28 and Equ. 4.29 is the termination step. On termination, this will give us the probability of the most probable state path in $O(N^2T)$ time. In order to retrieve the actual path one needs to maintain a table which will be used to record which value of j maximized Equ. 4.28. The path can then be retrieved by backtracking in the table [94].

Finally, the third and most difficult problem is essentially the problem of training HMMs with training data. No analytic algorithm exists for finding a solution to this problem and in [94] iterative techniques are suggested such as the Baum–Welch and the Expectation–Maximization (EM) algorithms which can give a local solution. What follows is an outline of the Baum–Welch procedure as presented in [94].

Assume that the observation distributions can be modelled by a Gaussian mixture model, i.e. $b_j(O) = \sum_{m=1}^{M} c_{jm}\mathcal{N}(O, \mu_{jm}, \Sigma_{jm})$, where j is the state index, M is the number of Gaussian components in the mixture model, c_{jm} is the weight of mixture component m for state j, and $\mathcal{N}$ is a Gaussian density distribution with mean μ_{jm} and covariance Σ_{jm}. Let us now introduce a backward variable $\beta_t(i) = P(O_{t+1}, O_{t+2}, ..., O_T|Q_t = S_i, \lambda)$ together with the following recursive equations:

$$\beta_T(i) = 1 \ , \tag{4.30}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \ , \tag{4.31}$$

where Equ. 4.30 is the initialization of the recursion and Equ. 4.31 is the induction step. Additionally, define two more variables ξ and γ as:

$$\xi_t(i,j) = P(Q_t = S_i, Q_{t+1} = S_j|O, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \ , \tag{4.32}$$

$$\gamma_t(i) = \sum_{i=1}^{N} \xi_t(i,j) \ . \tag{4.33}$$

With ξ and γ defined as above, $\sum_t \xi_t(i,j)$ represents the expected number of transitions from state S_i to S_j, while $\sum_t \gamma_t(i)$ represents the number of transitions leaving state S_i [94]. Finally, the following equations define the update rules of the iterative Baum–Welch procedure for the re-estimation of the HMM model parameters (which happen to coincide with the update rules of the EM algorithm for this problem) [94]:

$$\bar{\pi}_i = \gamma_1(i) \ , \tag{4.34}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \ , \tag{4.35}$$

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)} \ , \tag{4.36}$$

$$\bar{\mu}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) O_t}{\sum_{t=1}^{T} \gamma_t(j,m)} \ , \tag{4.37}$$

$$\bar{\Sigma}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)(O_t - \mu_{jm})(O_t - \mu_{jm})^T}{\sum_{t=1}^{T} \gamma_t(j,m)} \ , \tag{4.38}$$

where $\gamma_t(j,k)$ is the probability of being in state S_j at time t and the k^{th} mixture component accounting for the observation O_t. In particular:

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jk}\mathcal{N}(O_t, \mu_{jk}, \Sigma_{jk})}{\sum_{m=1}^{M} c_{jm}\mathcal{N}(O_t, \mu_{jm}, \Sigma_{jm})} \right] \tag{4.39}$$

Repeated use of this iterative procedure converges to a local optimum. Note that training requires a large number of data in order to achieve a good solution.

4.4.3 Overview of Spectral Clustering

On one hand, by combining appearance (PSIFT) and shape features (PHOG) improves the discriminative power of our representation, it increases the dimensionality of our input. As such, it increases the amount of training data that we need in order to learn accurate HMM models, and this also causes an increase in complexity, thus slowing down computations.

Spectral clustering [85] is a popular method of dimensionality reduction. The feature vector of each training example is represented as a node in a graph that is connected with a weighted edge to its nearest neighbors in the training set (weights reflect degree of similarity between training examples). The algorithm then applies an eigenvalue decomposition on the matrix representing this graph, reducing the feature vector dimensionality in a way that preserves the neighborhood structure. We use SPMK as the affinity measure and reduce the dimension of PSIFT and PHOG features separately. Figure 4.12 shows the resulting embedding of the training set, where we see that the classes are well separated. We then apply the embedding to the test set [4].

The final feature descriptors per frame are the combined SIFT and HOG features of reduced dimensionality, together with the 3D head pose and its first order derivatives. These are then used to train a separate HMM for each class, using sequences segmented by class. In order to do continuous recognition, we slide a window over each test sequence, classifying the frames within it as negative, topic, wh or none, based on

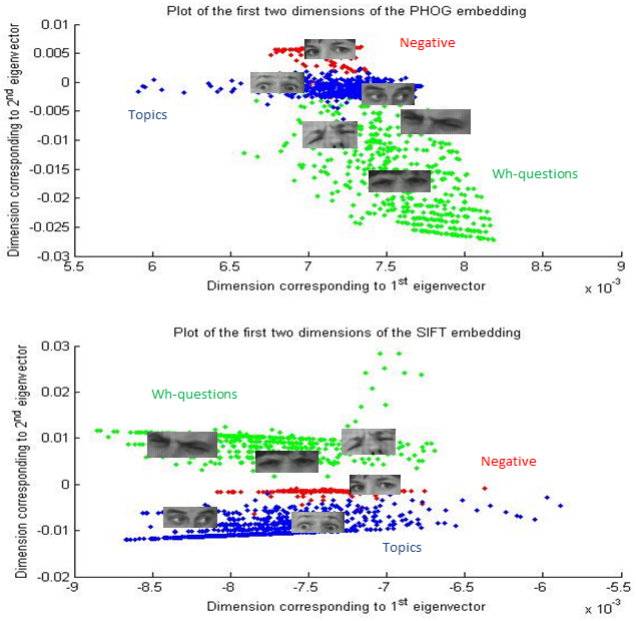

Figure 4.12: Spectral feature embedding of each frame (red: negative, green: topics, blue: wh-questions).

which HMM yields the highest probability of having generated that subsequence.

4.4.4 Experimental Results

From the dataset of the Boston University American Sign Language Linguistic Research Project (ASLLRP), we selected a training set of 77 video clips of isolated utterances (negative: 17, topic: 40, wh: 20). Our testing set contained 70 such clips (negative: 15, topic: 38, wh: 17). The exact composition of these sets, in terms of numbers of frames per class, is shown in Table 4.10. Both sets contained three different signers. Using the methods described in previous sections [76], we tracked the signer's head, extracting pose, PHOG and PSIFT features, the dimensionality of which was then reduced using spectral clustering. We then trained class-specific HMMs, optimized to recognize frame sequences of their class. To evaluate their performance at continuous recognition, we used a sliding window approach. We fed sub-sequences of all unsegmented test sequences to each HMM, classifying each frame as negative, topic, wh, or none, based on

	None	Negative	Topic	Wh-Q
Training	10144	997	1604	1208
Testing	9359	1053	1248	1182

Table 4.10: Dataset composition (number of frames per class).

	Predicted Class			
	None	Negative	Topic	Wh-Q
True None	92.8%	2.9%	2.2%	2.1%
True Negative	7.7%	80.3%	5.8%	6.2%
True Topic	9.2%	4.5%	81.2%	5.1%
True Wh-Q	8.3%	5.3%	4.5%	81.9%

Table 4.11: Confusion matrix of HMM continuous recognition.

which HMM output had the highest probability of having generated each subsequence. Recognition accuracy is summarized in the confusion matrix of Table 4.11.

4.5 Head Pose Normalization

When the head rotates, there is a variation in facial appearance when viewed from a fixed point, e.g., a static camera. For a difficult task, such as that of continuous recognition of non-manual markers in ASL, where many classes differ only in subtle ways from each other, this variation in appearance further complicates the recognition problem. As a result, more training data is needed to correctly learn the facial appearance manifold and more complex algorithms to model the sequential appearance transitions for the purpose of non-manual marker recognition. In this section we present an extension to our framework, which aims to address this problem.

More specifically, our extended method involves combining the 2D deformable face model, used by our extended face tracker, with a 3D deformable face model. The combined model enables correction for the warped appearance of faces due to variations in the 3D head pose, thereby leading to pose invariant facial features that in turn lead to improved recognition rates. We do this by registering landmarks on the 2D image, as predicted by our novel face tracker, with 3D landmarks on a trained statistical 3D

face model [5] to estimate the 3D projection matrix, R, (see Eq. 4.40). This matrix is then used to rotate the aligned 3D model to a frontal pose and re-project it to the 2D image plane, obtaining warped 2D landmarks. By estimating the difference between the tracked 2D landmarks and the warped 2D landmarks, we determine the image flow necessary to warp the input facial region to a frontal pose. The warped region is used to extract LBP features which we represent in a spatial pyramid. Together with an estimation of the eyebrow height (using the warped landmarks) and the signer's head pose, we use an HMSVM model for continuous recognition of wh-questions, negative sentences, conditional/when clauses, yes-no questions, topics and rhetorical questions. In the next sections we explain the 3D model in more detail and then present our results.

4.5.1 3D Face Model

Let the shape vector $s = (x_1, y_1, z_1, \cdots, x_n, y_n, z_n)^T$, which contains X, Y, Z coordinates of its n vertices, represent the 3D geometry of a face, and define a morphable face model using Principal Component Analysis (PCA) on the training dataset. Following Yang et al.'s work [128], we fit the 3D model to a face image by solving for the projection matrix R, using a two-step Least Squares minimization: (1) estimate the deformation parameters of the statistical 3D face model that best match the 2D landmarks when projected on the image, and (2) estimate the projective transformation (which includes scaling, rotation and translation) that maps the aligned 3D landmarks to the 2D image. Steps (1) and (2) are repeated until convergence. In other words, we minimize the projections of 3D landmarks X_k, and the corresponding 2D landmarks Y_k tracked by the ASM face tracker using:

$$min \sum_k ||Y_k - R \cdot X_k||^2 \ . \tag{4.40}$$

The projection matrix R defines the scale, rotation and translation of the 3D shape. If we change the rotation angles to zero, we will get a new projection matrix which projects the 3D shape to a frontal pose, but with same scale and translation. By

	Cond/When	Negative	Topic	Wh-Q	Yes-No	Rhetorical
Training	31	33	22	35	28	26
Testing	20	23	15	27	19	18
Total	51	56	37	62	47	44

Table 4.12: Dataset composition (number of utterances per class).

comparing the difference between the two projections, we can compute the movement of each vertex i on the image plane using:

$$F_i = R_0 \cdot X_i - R \cdot X_i \ , \tag{4.41}$$

thereby warping the face region to a frontal pose.

4.5.2 Experimental Results

From the dataset of the Boston University American Sign Language Linguistic Research Project (ASLLRP), we selected utterances containing conditional/when clauses, negative sentences, wh-questions, yes-no questions, topics and rhetorical questions. We used about 60% of the utterances for training and the rest were used for testing. The detailed composition of the dataset used is shown in Table 4.12. In total, there were 24717 frames of video.

Using the methods described in previous sections, we tracked the signer's head and then applied the 3D face model on each input video frame. The 3D model was trained by PCA using range data from 6 different subjects performing 5 different facial expressions in 5 sessions [5]. In each frame, we used the tracked 2D landmarks to estimate R, which we used to estimate the image flow needed to warp the face region to frontal pose. The warping procedure was implemented in MATLAB and it took about 1 second per frame, while face tracking was at a rate of about 6 fps as in other experiments (Intel Quad Code 2.4GHz, 8GB RAM). Figure 4.13 illustrates the image warping process.

From the face ROI of the warped input image, we extracted LBP features which we encoded to spatial pyramids of $L = 2$ levels and then reduced the dimensionality using

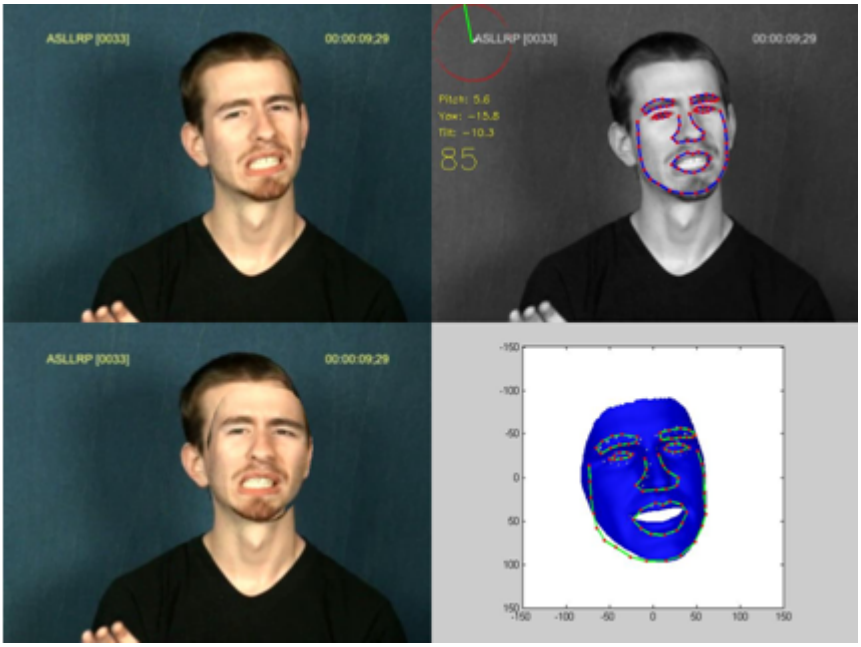

Figure 4.13: Illustration of process for warping facial region to a frontal pose. (Top left) Input frame; (Top right) Tracked result; (Bottom right) Registration of 2D landmarks to 3D face model; (Bottom left) Resulting warped facial region.

PCA to preserve 95% variance. We used the warped 2D landmarks to estimate eyebrow height and eye distance, which together with the head pose were augmented with the LBP features to get the final feature descriptor. We then trained a multi-class HMSVM model with a linear kernel. Parameter C was determined by 5-fold cross validation and the mean confusion matrix is summarized in Table 4.13. We repeated the experiment but this time the warping stage was skipped, so as to evaluate the effect of warping on the classification rate. The box plot in Fig. 4.14 illustrates that on average ($N = 5$ runs) the accuracy with the warping improves by about 2.5 units, i.e., from 81% to 83.5%. The classification performance is very good for the none, the conditional/when clauses, the wh-questions and the yes-no questions, and the confusions are reasonable in most cases, e.g., rhetorical and yes-no questions both involve raised eyebrows. Examination of the predictions per sequence revealed that most confusions happened at the beginning or the end of a marker. This suggests that more discriminative features must be needed.

Nevertheless, use of the 3D model did improve the overall performance. The benefit

	Predicted Class						
	None	C/W	Neg	Top	Wh-Q	Y/N	Rhet
True None	90.0%	1.6%	3.1%	1.1%	2.6%	1.2%	0.4%
True C/W	1.8%	85.3%	0.3%	1.7%	0%	10.9%	0%
True Neg	41.7%	0%	53.6%	0%	4.6%	0%	0.1%
True Top	20.2%	13.0%	0%	55.1%	0.3%	0.7%	10.6%
True Wh-Q	2.9%	0%	0%	0%	97.1%	0%	0%
True Y/N	2.2%	9.7%	0.3%	2.5%	0%	84.5%	0.8%
True Rhet	21.8%	5.2%	10.3%	1.6%	0.5%	7.8%	52.9%

Table 4.13: Confusion matrix of HSVM continuous recognition with head pose normalization.

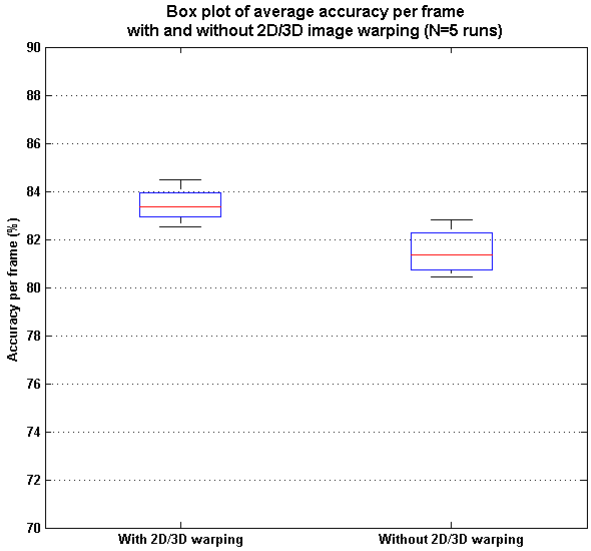

Figure 4.14: Box plot of average accuracy per frame with and without 2D/3D warping for N=5 runs. Warping consistently improves the classification accuracy.

of this transformation of the face region to a frontal pose is that it filters out the effects of head orientation from the appearance features (histograms of Local Binary Patterns) that we compute from the given region. This means that training of the facial appearance models does not rely on the head pose of the subjects (although we still use head pose information as a separate feature channel), but only on the actual facial expression, e.g., eyebrow configuration. Therefore, we can learn pose-independent recognition models and with less training data because there is less variation in facial appearance (i.e., we only observe variation due to changes in facial expressions) than when the head is allowed to rotate freely in 3D. In addition, this method filters out the effects of foreshortening on the estimation of the eyebrow height, e.g., when the eyebrow-eye distance appears to be shorter when the head is looking high up. Future work will be geared towards augmenting the 3D face model with a texture model for more accurate warping from larger rotations to frontal pose. We also plan to work towards a hierarchical learning model, where we explicitly model and recognize head shakes, eyebrow raises, eye widening, etc., and then fusing these features at a meta-level, together with features from the manual component of the ASL signal, for better detection of marker boundaries and better classification.

4.6 Summary

In this chapter we have applied our extended face tracker to the problem of non-manual marker recognition in ASL video. We first presented our framework for the relatively easier task of isolated recognition and presented an extension which handles feature misalignment and models temporal feature dynamics, allowing us to improve our classification performance and also expand the number of different classes of non-manual markers that we can recognize. Then we described our method for the harder task of continuous recognition, which we later extended with a 3D face model for head pose normalization. Our methods were validated with experimental results.

Chapter 5

Deception Detection from Kinesic Analysis

In this chapter, we describe the application of our face tracking system to the problem of deception detection from kinesic analysis [16, 74]. In particular, we focus on deception detection in a subject's responses to a set of interview questions, using an un-calibrated camera to record the subject's body movements and facial expressions throughout the interview. The first method we present builds subject-specific models to detect deceptive over-control and agitation, while our second method is a proof of concept for deception detection via synchrony analysis of the kinesic behavior of a subject and their interviewer.

5.1 Theoretical Background

We begin with a brief theoretical background on the psychology of deception in order to justify our approach and lead to our hypothesis relevant to this chapter. In situation where the stakes are high liars exhibit higher cognitive load, stress, arousal and negative emotions than truth-tellers [16, 29, 35, 90, 137], since they are trying harder to make themselves believable and conceal their deception. The higher cognitive load may cause a decrease in overall movement if the liar's brain gets preoccupied with story fabrication and deception concealment, thereby leaving fewer cognitive resources for performing other body movements. The arousal and nervousness may cause an increase in fidgeting and self-adaptor gestures (e.g., touching one's face). Since their goal is to conceal their deception and appear believable, liars attempt to suppress their increased negative emotions by exercising over-control or try to mask them by smiling, for example. Moreover, general expressive gestures (i.e. illustrators) may decrease during deception, as the liars attempt to exercise over-control, except for adaptor movements

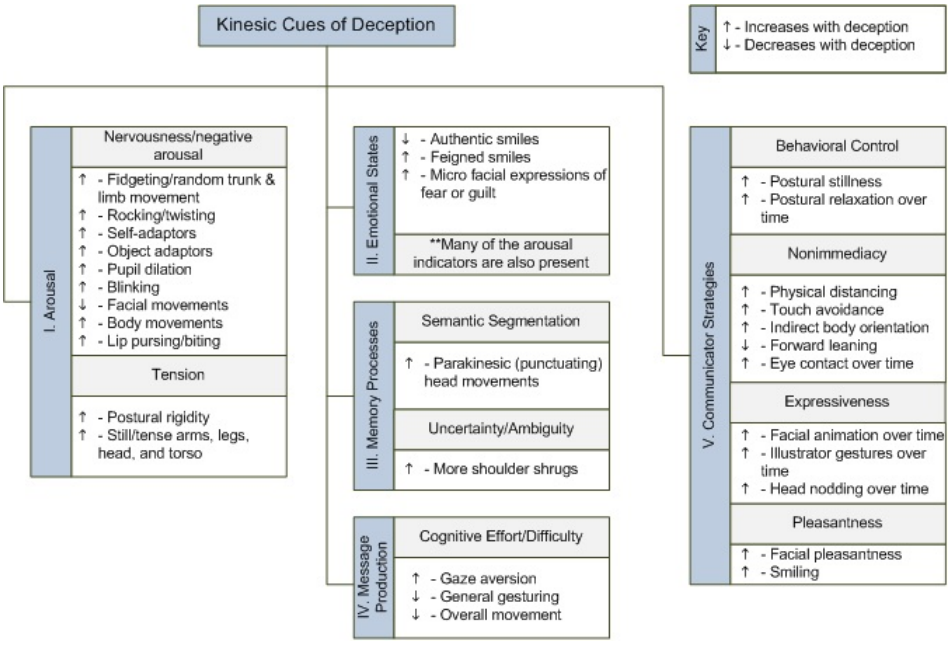

Figure 5.1: Diagram illustrating a summary of kinesic cues and how they are affected by deception [15, 25]

mentioned above, which may increase as a result of their nervousness. A summary of kinesic cues and how they are affected by deception is shown in Fig. 5.1.

Furthermore, some deceptive behaviors fall into one of two groups. The first one is that of "over-control" and the other one is that of "agitation" [34]. Over-controlled liars, as they attempt to conceal their deceit, particularly those who may be aware of possible behavioral cues that signal deception, they may try harder to hide all potential behavioral indicators. For example, they may reduce the movements of their hands, legs and head [11, 25, 122, 123]. Agitated liars are at the other extreme. The fear and nervousness brought on by their deception causes their speech to become faster and louder and they may show an increase in fidgeting [25, 74]. This categorization agrees with the literature that suggests an increase in adaptors and a decrease in illustrators during deception.

Nevertheless, there is no single behavioral or psychological cue that is a perfect indicator of deception [16]. The various cues must be judged in their context. For example, a subject may make more arm movements if they are standing as opposed to

when they are comfortably resting their arms on an armchair. In addition, a subject's "normal" behavior as well as the tone and context of the conversation are equally very important. For example, a subject may incorrectly behave as being over-controlled because they may tend to get nervous when interrogated, regardless of whether they are lying or not. Similarly, a subject may show agitated behavior if they are questioned soon after they had exercised. Therefore, it is important to establish, or otherwise model, a baseline behavior, in order to improve the accuracy of deception detection.

This need is reiterated in the Expectations Violation Theory (EVT) of Burgoon [14], which states that if in a communication there is significant deviation of the expected behavior from what is actually observed then this should cause suspicion. In other words, if a subject is calm until they are asked about the theft, at which time they deny it and start fidgeting excessively, then clearly the subject may not be telling the whole truth. Additionally, the Interpersonal Deception Theory (IDT) of Buller and Burgoon [11] states that deception is a dynamic process, whereby the deceiver adjusts their behavior according to how much they thing they are believed, which means that their behavior is also dependent on the person asking the questions, and this is another reason why session calibration even for the same subject may be necessary. Regardless, deceivers will at some point involuntarily reveal some behavioral cue as a result of their deception and suspicion [73] making deception detection via kinesic analysis possible.

Therefore, we hypothesize the following. We expect that deceivers may reduce their overall gestural movements (illustrators) and may increase their adaptor movements for the reasons already mentioned. Thus, to distinguish truth from deception it will suffice to implicitly classify a subject's behavior into normal, over-controlled or agitated. Because of the dynamic nature of deception and the variation in the population, we will train subject-specific models, which will establish a different baseline behavioral profile for different subjects.

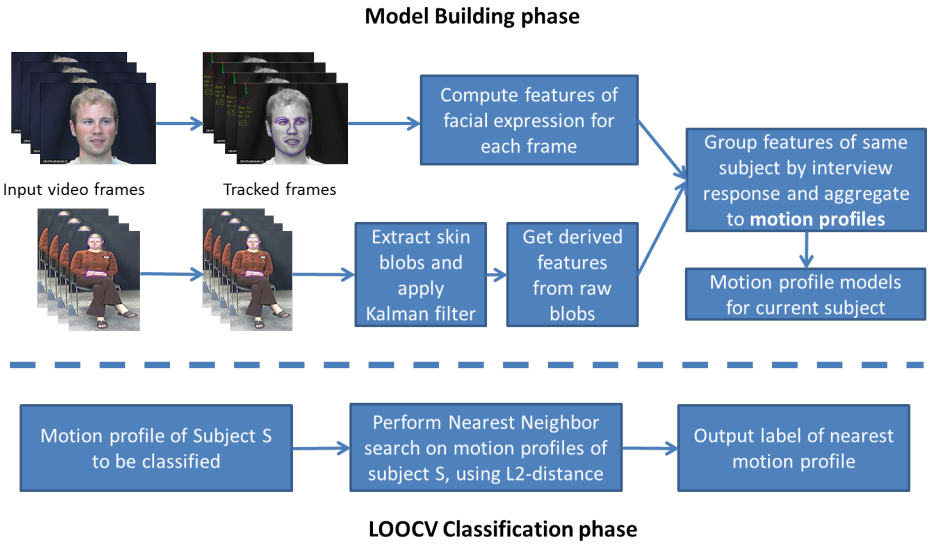

Figure 5.2: Overview of our kinesic analysis. In the model building phase the input video is tracked with the face tracker and the skin blob tracker, extracting features which are then grouped by interview response and aggregated into motion profiles. During Leave One Out Cross Validation (LOOCV) classification, the nearest neighbor's class label is used to classify interview responses as deceptive or truthful.

5.2 Methodology

Kinesic analysis requires accurate tracking of the target's body movements (e.g., hand gestures, head movements, shoulder shrugs) and facial expressions (e.g., eyebrow movements). Once the target is tracked, various features, which can serve as behavioral indicators of deception, are computed from the output of the tracking module. Therefore, we use the face tracker we described in Chap. 3 to track facial features and we adopt a modified version of the skin blob tracker used by Lu et al. [70] to track gross movements of the hands and head for every frame of an interview video. Note, however, that more sophisticated methods for adaptive skin blob tracking can be used, such as the Markov model of Sigal et al. [101] or the Gaussian Mixture Model presented in [135]. For our dataset we found that the simpler and computationally more efficient method of Lu et al. [70] sufficed. For each video frame, the face tracker outputs the (x,y) image coordinates for each tracked facial landmark and the 3D head pose, while the skin blob tracker outputs the (x,y) image coordinates, orientation and size of each

detected skin blob (short and long axis of ellipse and ellipse area). These raw features are used to compute derived features (see Sec. 5.3) to characterize postural changes, frequency of adaptor movements, frequency of illustrator movements, head gestures, etc. Features are then grouped, so that all features extracted from frames when the subject was responding to a particular interview question are aggregated together, to form motion profiles, one for each response to an interview question. Classification of an interview response as truthful or deceptive is done by comparing the motion profile of that response to those from the other responses of the same subject, assigning to it the label of its nearest neighbor in the L2-norm sense. The overall procedure is summarized in Fig. 5.2. Next we give an overview of the skin blob tracker, which we use to track regions of skin color (head and hands).

5.2.1 Skin Blob Tracking

Following the method in [70], we use color segmentation to track the position, size and orientation of the head and hand blobs. Instead of a 3D Look-Up-Table (LUT) [55], we build a 2D LUT with Hue and Saturation color components, based on the Hue-Saturation-Value (HSV) skin color distribution of the face and hands. We use 32 bins for each of the Hue and Saturation components [55]. The Value component of the HSV representation is not used, so as to make the representation more robust to illumination than the normalized Red-Green-Blue (RGB) color representation used in [70]. The LUT is built off-line from skin color samples (see Fig. 5.3). Similarly, we build off-line a non-skin color model to model the color distribution of non-skin color pixels, using non-skin and background images. A pixel with color c is then classified as skin, if the ratio of the probability of it being skin, $P(c|\text{skin})$, to the probability of it being non-skin, $P(c|\neg\text{skin})$, is greater than a threshold, Θ:

$$\frac{P(c|\text{skin})}{P(c|\neg\text{skin})} \geq \Theta \ . \tag{5.1}$$

Using the LUT and Eqn. 5.1, the system produces a binary image of candidate face and hand regions in each frame, which are processed with morphological operators

Figure 5.3: Illustration of skin samples used to build the skin 2D LUT [55, 70] for skin regions in Hue-Saturation color space. In a similar fashion we collect non-skin samples and build a 2D LUT for non-skin regions.

(open and close) to yield candidate elliptical head and hand blobs. Candidate blobs are pruned by taking into account the previous position and shape of each of the blobs [70]. Note that we may not get three distinct blobs in every frame. This may happen, if for example, the hands are touching each other or the head or if any of the hands are not visible. The system can detect blob merging, which indicates that the subject is performing an adaptor gesture, by checking the size of the blob area. For blobs that are not visible, we reuse their last known position, while it duplicates merged blobs to end up with three blobs. A sample frame illustrating the detected blobs and the skin color samples used in building the color model is depicted in Fig. 5.4.

For robustness to tracking noise, we filter each of the three computed skin blobs using a Kalman filter [56], assuming linear state dynamics with Gaussian noise, $\mathbf{w}$. The system state for each skin blob, $\mathbf{x}_t^i$, where i is one of {left-hand, right-hand, head}, includes its (x, y) position, its area, A, its orientation angle, θ, the length of its short, L_s, and long axis, L_l, and their first order derivatives, so that $\mathbf{x}_t^i = [x, y, A, \theta, L_s, L_l, \dot{x}, \dot{y}, \dot{A}, \dot{\theta}, \dot{L_s}, \dot{L_l}]^T$, at time t. The dynamic process is governed by:

$$\mathbf{x}_{t+1} = A_k \mathbf{x}_t + \mathbf{w}_t \ , \tag{5.2}$$

Figure 5.4: Sample frame showing the tracked head (blob 0) and hands (blobs 1 and 2) of an interviewee. The tracker records the (x,y) coordinates, area and axis lengths of each detected blob. The skin color samples are shown in the upper right corner.

with

$$A_{t+1} = \begin{pmatrix} 1 & \delta t \\ & \\ 0 & 1 \end{pmatrix} \text{ and } \mathbf{w}_t \sim \mathcal{N}(0, Q) \ . \tag{5.3}$$

The observation process is modelled as:

$$\mathbf{z}_t = H\mathbf{x}_t + \mathbf{u}_t \ , \tag{5.4}$$

where $H = [1, 0]$, $\mathbf{z}_t$ is the observation as obtained by the skin blob tracker and $\mathbf{u}_t \sim \mathcal{N}(0, R)$ is the observation noise at time t.

5.3 Feature Extraction

In every frame we track the movements of a subject's hands and head relative to their body (via the skin blob tracker), as well as some of their facial expressions and their 3D head pose (via the face tracker). In addition to these raw features, which we use in our feature descriptor, from the tracked positions of the blobs we compute derived features, as in [73], which are designed to capture behavioral agitation and over-control, by

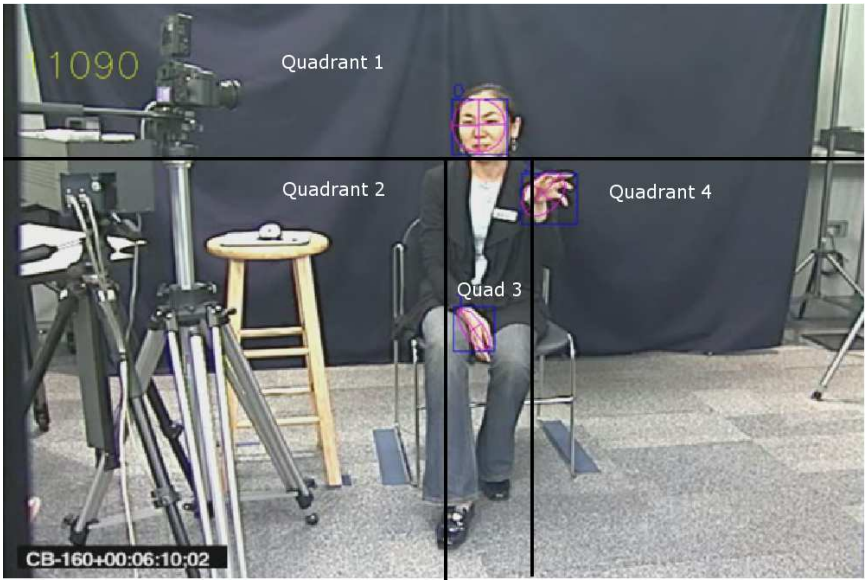

Figure 5.5: Illustration of quadrant features. They are used to capture the positions of a subject's hands relative to their body.

characterizing relative positioning of the hands, postural shifts and postural openness of a subject.

5.3.1 Skin Blob Features

We divide frames into quadrant regions and these are shown in Fig. 5.5. For each frame, we record in which quadrant each of the hand blobs is found, based on its (x,y) position. The trend of quadrant locations for each hand blob over a period of time can give us an idea of how frequently the subject is performing illustrative gestures or maintaining an open posture (e.g., quadrants 1, 2 and 4). On the other hand, when the hand blobs are consistently in quadrant 3 we can deduce that the subject is maintaining a closed posture, possibly a sign of over-control, or is making adaptor gestures (i.e., hands touching each other).

Furthermore, imagining that the hand and head blobs form the vertices of a virtual triangle, we can use the area and centroid of this triangle to quantify the openness of a subject's posture and any postural shifts [73]. The triangle area feature is shown in Fig. 5.6. Triangle area is expected to increase for open postures and while gesturing

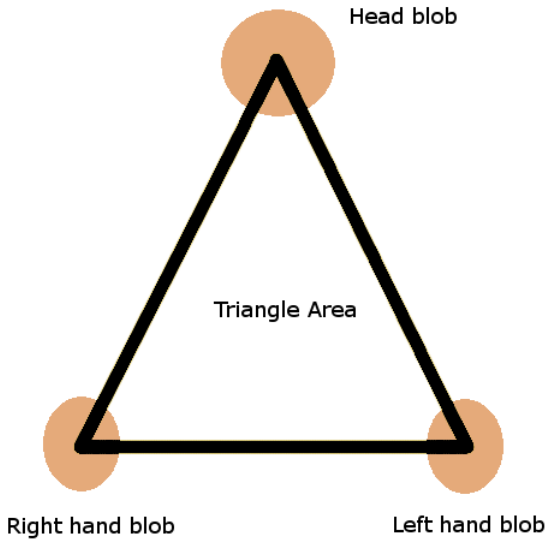

Figure 5.6: Illustration of triangle area feature. It is used to quantify the degree of posture openness of a subject.

and decrease for closed postures and over-controlled behaviors. Similarly, the triangle distance features (see Fig. 5.7) will increase for open postures and decrease for closed postures. These features can also indicate if a posture is symmetric or if the subject's body is leaning on either side (e.g., if the left hand distance to the triangle's center is shorter than that of the right hand, then the subject is leaning to the left). Lastly, the triangle angle features (see Fig. 5.8) also indicate relative orientation and positioning of the blobs and by looking at their change over time we can observe the gestural patterns of the subject, while similar values over a period of time signal that the subject is not gesturing much.

In order to account for differences in subject sizes and positioning, we also look at the change of these feature values over time. For example, we compute blob displacement $(\Delta x_{t_i}, \Delta y_{t_i})$ at time t_i, which is also proportional to velocity, using:

$$\Delta x_{t_i} = x_{t_i} - x_{t_{i-1}} \ , \tag{5.5}$$

$$\Delta y_{t_i} = y_{t_i} - y_{t_{i-1}} \ , \tag{5.6}$$

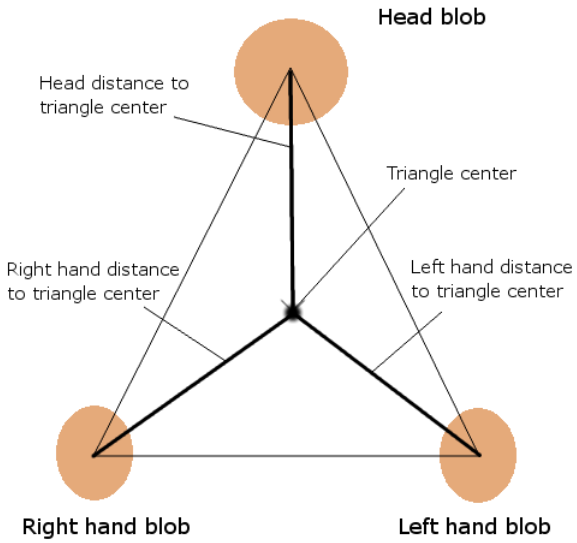

Figure 5.7: Illustration of distance features of each of the blobs (left hand, right hand and head) to the triangle's center.

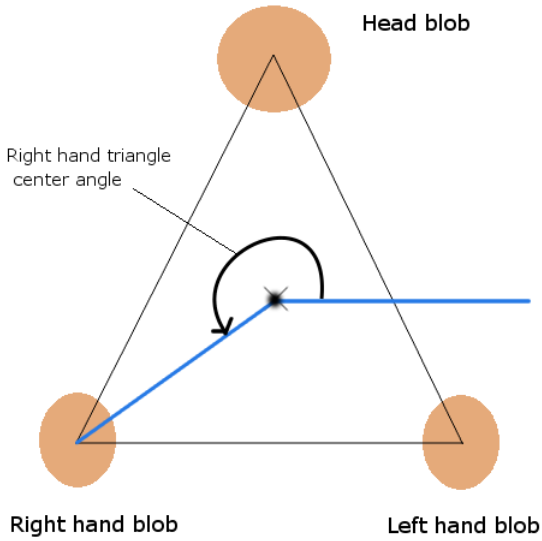

Figure 5.8: Illustration of angle features of the blobs relative to the triangle's center.

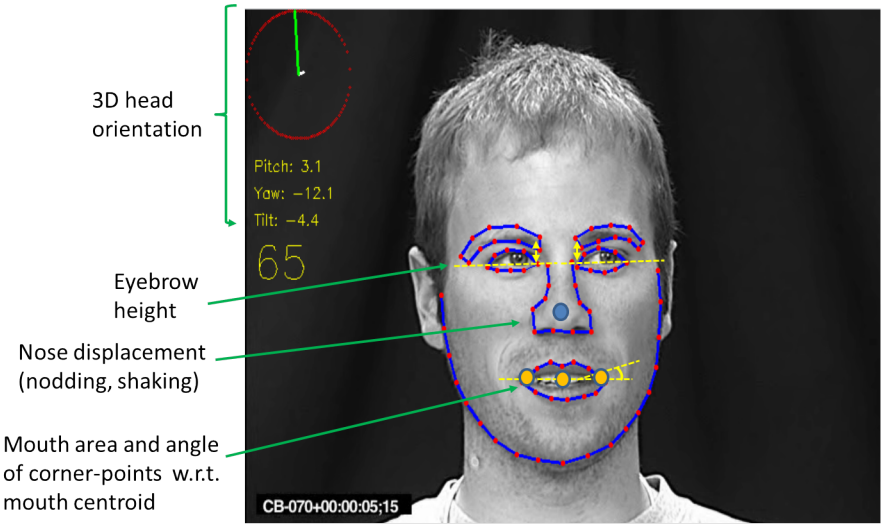

Figure 5.9: Sample tracked facial frame from kinesic analysis, illustrating the features derived from the face tracker's output: (a) 3D head angle and nose-tip displacement (displacement $\propto$ velocity) for detecting head nodding and shaking, as well as other head movements, (b) eyebrow height for catching eyebrow expressions (c) mouth area and orientation of mouth corners with respect to mouth centroid to capture mouth expressions.

where (x_{t_i}, y_{t_i}) is its position at time t_i.

5.3.2 Facial Features

Moreover, the 79 tracked landmarks and 3D head orientation angle (pitch, yaw, tilt) output by the face tracker, were used to compute derived features designed to capture facial micro expressions and asymmetries. Namely, these are the following: change in angle between the mouth's centroid and each of its corner points, change in mouth area, displacement of inner and outer left and right eyebrows. The left/right mouth corner points are computed as the means of the three leftmost/rightmost mouth landmarks. Finally, the displacement of the eyebrow (change in eyebrow height) is computed using the mean displacement of the four innermost eyebrow landmarks. From this displacement we subtract the mean displacement of the six lower nose landmarks (i.e., the position of the nose-tip) to account for head displacements, assuming that the nose is the most stable face component. The nose-tip displacement is used to also detect head

nods and head shakes (e.g., a decrease, then an increase and then another increase in x or y nose-tip displacement). The 3D head pose and its change over time provides complementary information for detecting head shakes (yaw angle) and head nods (pitch), as well as useful information about where the subject's head is pointing, relative to the interviewer, and whether the subject is engaged with the interviewer or if they are frequently turning away from them, possibly because of nervousness. These features are illustrated in the sample tracked frame of Fig. 5.9. Lastly, to our feature pool we append the normalized frequency of head nodding, head shaking and adaptors such as left hand touching the head, right hand touching the head, hands touching each other.

5.3.3 Non-parametric Descriptor: Motion Profiles

In order to summarize the tracked motions and expressions of subjects, while retaining as much information as possible about their distribution, we compute "motion profiles". These are similar to the movement signatures of [70], however our motion profiles include facial expression information as well. More importantly, and different from previous work [73], instead of using gross summary data in the form of means and variances of each feature channel, which causes loss of information about the fine distribution of feature values, we choose to use a richer non-parametric histogram representation. Additionally, we apply a log transformation to our motion profiles in order to capture information important to deception detection, namely, little or no movement (i.e., over-control), and extreme movement (i.e., agitation). In each subject's response the majority of frames involve a small amount of motion. In other words, subjects rarely make extreme movements for the entire duration of their response and use of regular binning will cause the bulk of the data to be crowded in the first few bins. Therefore, we change the scale of our data representation, in order to properly space out the data to enable discrimination of behavioral states.

More specifically, features are histogrammed into k uniform bins in log-feature space, meaning that each bin has an exponentially increasing size. Therefore, the first bin covers a very small range (corresponding to little or no motion) and the fifth bin covers

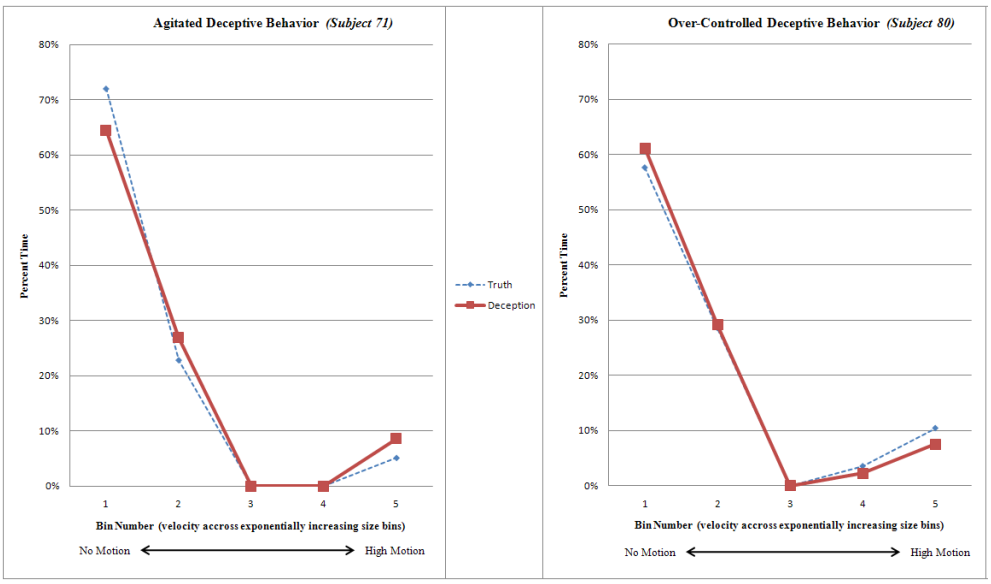

Figure 5.10: Average hand motion shown for two different subjects. Graphs show 5 velocity bins from "little motion" (leftmost) to "high motion" (rightmost). The subject on the left exhibits agitation when deceptive shown by an increase in the rightmost bin and a decrease in the leftmost bin of the red (deceptive) graph, relative to the blue (truthful) graph. On the other hand, the subject on the right exhibits over-control when deceptive shown by a decrease in the rightmost bin and an increase in the leftmost bin of the red (deceptive) graph, relative to the blue (truthful) graph.

the largest range (corresponding to all extreme motions). The bin range is subject-dependent as we set the bin log-range based on the feature range over the entire interview of a given subject, thus achieving subject normalization, which makes sense given that we are trying to distinguish deviations from a subject's baseline or normal behavioral patterns. Histograms for each channel are also normalized to have unit sum. This new representation of the data is successful at isolating the over-controlled and agitation responses that Ekman et al. [34] point to as being important indicators of deception. This is illustrated in Fig. 5.10, where we show the size of each bin for hand motion averaged over all responses for two different subjects. The graph on the left (using $k = 5$ bins) demonstrates a subject exhibiting agitated deceptive behavior: when responses are deceptive, the little motion bin shows a dip and the high motion bin shows a spike, relative to their truthful responses. The graph on the right demonstrates over-controlled deceptive behavior: when responses are deceptive, the little motion bin shows a spike and the high motion bin shows a drop, again, relative to their truthful

responses.

Let $\{x_{i,j}\}_{i=1}^{F}$ be the set of F features we extract from frame j as described in Secs. 5.3–5.3.3. By grouping together features extracted from m consecutive frames, we form a feature set of the form $\{\{x_{i,j}\}_{i=1}^{F}, \ldots, \{x_{i,j+m-1}\}_{i=1}^{F}\}$, which forms the basis of the motion profile over a response r_q of m frames. For each of the F feature channels, we compute a k–bin normalized log-scale histogram of the feature values $x_{i,j}$ for $j = 1, \ldots, m$, resulting in F histograms having a total of kF bins. We call $\vec{x}_{r_q}$ a motion profile because the histograms capture the distribution of feature values within the response. For classification we use a Nearest Neighbor classifier, whereby for a given test response, we assign it the label of its nearest training neighbor. The distance measure used is the L2-norm.

5.4 Experimental Setup and Results

A laboratory experiment was conducted at the University of Arizona during which interviewees, who represented diverse cultural backgrounds, responded to 13 one-word answer questions and 12 longer questions posed by experienced interviewers from the Defense Academy for Credibility Assessment. The short answer questions were mainly for the subjects to get comfortable with the environment. The longer questions followed the style of a polygraph examination and included the following kind of questions:

1. Autobiographical e.g., describe the worst job you ever had and explain why you disliked it.

2. Moral dilemma e.g., would you take credit for someone else's work and why.

3. Episodic memory e.g., describe a time you thought of stealing something.

4. Future oriented e.g., how will you spend your next vacation.

Interviewees answered some questions truthfully and some deceptively according to a predetermined sequence. Half were randomly assigned to follow a truth-first sequence and half were randomly assigned to begin with a deception-first sequence. Interviewers were blind to the truthful or deceptive sequence that each subject was instructed to

	Deceptive	Truthful	Total
Number of Responses	882	882	1764

Table 5.1: Data set composition showing number of deceptive and truthful responses (six of each kind per subject) used for Leave One Out Cross Validation. Numbers based on 147 subjects.

follow. During the interview, three digital cameras recorded kinesic behavior: one recorded full body profile view, one recorded full body frontal view and one recorded frontal facial view only. We run the skin blob tracker on the video from the full body frontal view and the face tracker on the video from the frontal facial view, while we did not use the profile view. After each block of three questions, interviewees rated their truthfulness in answering each question on a 0 (not at all) to 10 (completely truthful) scale. Interviews were typically 20–30 minutes long.

The recruitment efforts netted a multi-culturally diverse sample of 220 participants. Demographically, the mean age for the total sample was 28.9 years (while $\sigma = 13.34$), with 36% aged 21 and under, 48% aged 22 to 40, and 16% over 40 years of age. By gender, 55% were male and 45% were female. Subjects were given $10 to participate and another $10 if found credible, plus they were offered job interviewing tip. This scheme served as an incentive for them to make an effort to be perceived as credible by their interviewer for both their truthful and their deceptive responses, thus allowing leakage of deceptive behaviors that our method could capture. We selected the 147 subjects with an average self-rating over 7. We focused on the responses to the 12 longer questions in each interview, meaning that in total we had 1764 responses (half deceptive and half truthful). The data set composition in terms of number of frames involving deceptive and truthful responses is shown in Table 5.1.

Each video interview was analyzed and features were extracted from each frame. The full body frontal view was analyzed by the blob tracker and the facial frontal view was analyzed by the ASM face tracker, while the profile view was not used in our current analysis. We used 5 histogram bins (number determined by cross validation) per feature channel with uniform log space width (specific to the current subject). In

Method	Precision	Recall	Accuracy
Mock Theft Experiment [73]	59.2%	63.6%	60.0%
SVM	68.0%	70.1%	68.5%
Nearest Neighbor	**81.7%**	**81.5%**	**81.6%**
Nearest Neighbor with blob features only	78.2%	77.8%	78.2%
Nearest Neighbor with facial features only	69.2%	69.4%	69.3%

Table 5.2: Comparison of LOOCV classification accuracy rates for $N = 147$ subjects. Accuracy refers to the percentage of correctly classified responses. Precision and recall rates are for deception detection.

this way, the first two bins were wide enough to capture the very small feature values corresponding to over-controlled behavior, while the width of each of the remaining bins was successively increased to capture increasingly larger movements corresponding to relaxed and agitated behaviors, respectively. We built 147 separate Nearest Neighbor models (one for each of the 147 subjects), using Leave One Out Cross Validation (LOOCV), where for each of the interview responses, we hold one out to be used for testing and train the model on the rest, reporting the average LOOCV performance over all 147 NN models. We also tried an SVM classifier [113] for each of these 147 subject-specific models with an RBF kernel (scale and complexity parameters determined by cross validation for each subject). All results are shown in Table 5.2, while Table 5.3 shows the mean confusion matrix from all NN models. Our motion profile NN models achieve an accuracy of **81.6%**, which is significantly better than the accuracy of the method in [73] for a similar interview scenario, and ours is over a larger dataset, too. However, note that, unlike our work, the authors of [73] attempt to build models that generalize over all subjects, and are, thus, doing LOOCV per subject. Instead, we build 147 subject-specific models, doing LOOCV per response per subject. It is, therefore, clear that our subject-specific models perform better than a general model over all subjects. We attribute this to the fact that different subjects may have different deceptive behaviors and different baseline truthful behaviors, as opposed to there being a universal deception cue or "threshold", which holds for everyone and discriminates truth from deception, as has been hypothesized, for example, by Meservy et al. [72, 73].

NN Conf. Matrix	Pred. Deceptive	Pred. Truthful
True Deceptive	81.5%	18.5%
True Truthful	18.3%	81.7%

Table 5.3: Mean confusion matrix of Nearest Neighbor classifiers averaged over $N = 147$ subjects.

Our proposed system as presented has the limitation that training data must first be collected for a test subject so that the model can be trained. Acquiring such training data might not be trivial in the situations where such a system can be useful, e.g., at rapid screening checkpoints. Nevertheless, the proposed work does achieve state of the art performance using only visual cues in an automatic, covert system, and can serve as the foundation for understanding exactly what constitutes the peculiarities that characterize the deceptive tactics of different individuals.

5.4.1 Running-time analysis

Our experiments were conducted on an Intel Quad Core processor clocked at 2.4GHz with 8GB of RAM. Both trackers were implemented in C++ with OpenCV libraries. Video resolution was 720 x 480. The face tracker processed the videos off-line at about 6 fps ($\sim$ 170ms per frame) using 40 particles for the sample set. The skin blob tracker processed videos at a rate of 5 fps ($\sim$ 200ms per frame). The tracking implementations can be optimized with GPU programming to run in real-time, if desirable by the application. For our purposes, it was not necessary, given that all processing was done off-line and the evaluation method was LOOCV. Derived features were extracted from the trackers' raw output using MATLAB code which run significantly faster with a linear complexity O(n): linear time for the first pass through the data of a given subject to derive the features marking the min and max value of each feature (which serves to find the histogram bin ranges for the motion profile) and linear time for the second pass which builds the motion profile. In fact, the bottleneck here was just the I/O for reading text files and parsing the raw features. LOOCV classification by Nearest Neighbor takes about 40 seconds for the whole dataset (147 subjects), which is about

0.27 seconds per subject and 0.02 seconds per response on average, while LOOCV classification using SVM models (RBF kernel) takes about 80 seconds, which is about 0.54 seconds per subject and 0.05 seconds per response on average.

5.5 Method Extension: Subject-Interviewer Synchrony Analysis

Here we present an extension to our deception detection method presented in preceding sections of this chapter. However, instead of focusing our kinesic analysis on only the subject in the interview scenario, we make first steps in analyzing the behavioral patterns of both the subject and their interviewer, as they are interacting together. Besides, the Interpersonal Deception Theory (IDT) of Buller and Burgoon [11] states that deception is a dynamic process, whereby the deceiver's behavior and overall "strategy" is influenced by the actions of the agent they are interacting with. Therefore, we hypothesize that it is possible to detect deception in interview scenarios by joint tracking and analysis of the behaviors of both the subject and their interviewer. We validate our hypothesis with experimental results.

5.5.1 Theoretical Background

Human interactions are complex and multidimensional, they have verbal and non-verbal components. We are just scraping the tip of the iceberg by looking at the verbal behaviors of the subject through our kinesic analysis method, described earlier in this chapter. Besides, any human interaction involves at least two parties: a subject and an interviewer (or any otherwise interacting agent). Are we neglecting important information by focusing on just the subject of an interview? Practitioners in the field of deception detection and psychologists are suggesting that this is indeed the case, and that rapport-building techniques as well as analysis of the synchrony between a subject and their interviewer may serve as an effective method for detecting deception with terrorists [111], in FBI interviews [81] and in police investigations [58].

Two people interacting are said to be in synchrony when one person is mirroring or complementing the behaviors and gestures of the other, when there is a visible

coordination and matching in their actions. For example, person A smiles as person B is talking and then soon later person B also smiles to match that facial expression. The hypothesis maintained by these practitioners is that deceivers are less in synchrony with their interviewer, who overwhelmed with the stress and cognitive load of their deception fail to recognize and keep up with synchrony cues, unlike truth-tellers who find it easier to stay in synchrony.

In general, synchrony can be manifested in a number of verbal and non-verbal ways:

- Postural and gestural behaviors (e.g., head nodding)
- Movement
- Gaze
- Facial affect (e.g., smiling)
- Accents
- Speech rate
- Vocal intensity
- Self-disclosure

Non-verbal manifestations of synchrony can be detected by our face tracker, provided we have synchronized video feed for both the interviewer and the subject. Given the tracked landmarks, our system can observe, for example, variations in nose-tip displacement to detect head nodding and head shaking. In order to be able to detect facial expressions, we have implemented the method of Yang et al. [130] which is based on AdaBoost models of encoded Haar features extracted from a cropped facial region. The region is cropped based on the position of the eyes and nose-tip, which is provided by our extended face tracker. For purposes of proof-of-concept, we were only interested in detecting smiling events, so we only trained a model for smiling and one for no-smiling, augmenting it into our face tracker system. This new module takes as input the current frame and the tracked position of the landmarks. It uses the landmarks to crop and

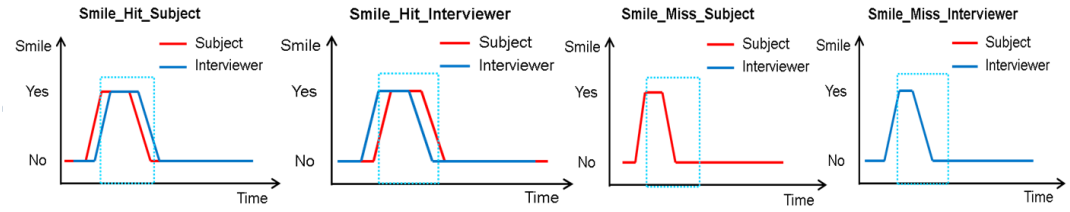

Figure 5.11: Illustration of the definitions of hit and miss events by subject and interviewer, which are used in our synchrony analysis for the detection of deception.

align the face region in terms of the eyes and nose-tip and then computes Haar features which it uses with the trained model to output a smiling and a no-smiling score. If the smiling score is greater than the no-smiling score and also above the threshold score of 0.5, it outputs the detection of a smiling event.

5.5.2 Quantifying Degree of Synchrony

Our system can detect timestamped dynamic events (as is or with the mentioned extensions, i.e., [130] for expression recognition), yet these raw events are not informative about the synchrony between a subject and an interviewer. Synchrony is at least a dyadic interaction, so we must look at the event detections for both parties together. Therefore, once the system has processed the videos of both subject and interviewer, recording any detected events (in this case we focused on head nodding, head shaking and smiling events), another module parses the detections for both streams, using a sliding window of size w, to output synchrony meta-events. If event A, where $A \in \{\text{nodding}, \text{shaking}, \text{smiling}\}$, is detected for subject X at time t and for interviewer Y at time $t \pm w$, where w is the size of the sliding window, we say we have a synchrony hit for that event, otherwise we say we have a synchrony miss for that event. In the case of a hit, we say it is a "subject hit", if the subject first performed the event and the interviewer followed and mirrored it. We say it is an "interviewer hit", if the interviewer first performed the event and the subject then followed and mirrored it. The definitions of hit and miss smiling events are illustrated in Fig. 5.11. Hit and miss events for head nodding and head shaking gestures follow a similar definition. The complexity of the sliding window algorithm for parsing synchrony events, which depends on the length of

the streams and on the window size, is $O(nw)$.

5.5.3 Experimental Setup and Results

In order to validate our hypothesis we have analyzed stimulus videos collected from an experiment at Oklahoma University (OU), in which Federal Polygraph Examiners interviewed subjects (N = 243) who cheated in a task. Each video file contained synchronized twin-view, showing both the subject and the interviewer. The subjects were OU students, while there were four professional polygraph examiners, provided by the Department of Defense, each having 4–30 years of experience.

More specifically, the participants were asked to play a trivia game with a confederate who gets them to cheat. These cheaters, who henceforth we refer to as the deceivers, are immediately afterwards interviewed by a professional examiner, while both being videotaped. The interviews were structured to have three segments and in each segment a different type of questions were asked. In the first segment the examiner is asking "background" questions, such as "what do you think is the purpose of this interview?" or "how well do you think you did at your task?" In the second segment they are asked "suspicion" questions, such as "are you aware of the prohibition against cheating in OU's honor code?" In the third and last segment, the subjects are asked direct accusation questions, such as "did you cheat?" or "why should I believe you?" In every segment the examiners were allowed to ask follow-up questions. The interview was structured in this way so as to initially establish trust between the subject and the interviewer. If towards the end of the interview, the synchrony between them broke down it would be a sign of deception, given the hypothesis that deceivers are less in-synch than truth-tellers. In fact, some of the subjects confessed to cheating, before the end of the interview, while others did not. For our analysis, we selected $N = 31$ subjects who did not confess.

We run our extended ASM face tracker on the selected videos to track the faces of subjects and their interviewer, while detecting and time-stamping gestures and facial expressions. In particular, we were interested in recording smiles, head nods and head shakes. The running-time was dominated by the facial expression recognition module

Figure 5.12: Simultaneous tracking of subject (left) and interviewer (right) illustrating detection of "head shaking" events (interviewer's face deliberately masked to protect identity). Only subject's head pose is shown. Subject did a head shake first, then the interviewer followed after about 1 second (37 frames of 30 fps video), so this is a subject-hit-shaking event.

Figure 5.13: Simultaneous tracking of subject (left) and interviewer (right) illustrating detection of "head nod" events (interviewer's face deliberately masked to protect identity).

which made the face tracking and event detection algorithm run at 5 fps on average (Intel Quad Core 2.4GHz, 8GB RAM) for a video resolution of 640×480. Figure 5.12 illustrates simultaneous tracking of subject and interviewer and detection of head shaking events, while Fig. 5.13 illustrates detection of head nodding events. In both cases, the face of the examiners is masked in order to protect their identity, given they are still active field agents. We then quantified the degree of synchrony in each video for each of the three interview segments, computing the "hit" and "miss" features discussed in Sec. 5.5.2, by running a sliding window classifier of size $w = 60$ frames, which is 2 seconds for a 30 fps video. The size of the window was determined by 10-fold cross validation. Figure 5.14 illustrates detection of synchrony features for head nodding events extracted from a sample video.

However, exactly because dyadic interactions are dynamic processes and individual behaviors also vary by person and by the context of the interaction it is imperative to normalize the synchrony features. This normalization should account for both the

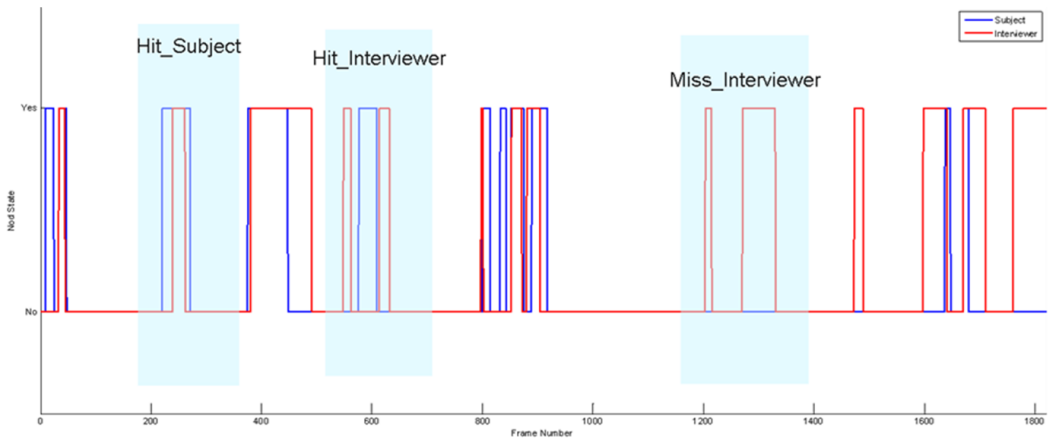

Figure 5.14: Illustration of hit and miss events for nod detection, which are used in our synchrony analysis for deception detection.

Classifier	Accuracy	TP Truthful	FP Truthful	TP Decept.	FP Decept.
Naive Bayes	58.1%	53.3%	37.5%	62.5%	46.7%
SVM	**80.6%**	**86.7%**	**25.0%**	**75.0%**	**13.3%**
Logistic	74.2%	73.3%	25.0%	75.0%	26.7%

Table 5.4: Deception detection rates using synchrony analysis and a 2-second synchrony window ($N = 31$ subjects). Accuracy is measured as the percent of correctly classified instances.

length of the interview segments and the amount of overall facial expressions and gestures that were produced by the interacting agents, so that it will allow implicit detection of out-of-synch interactions, which in turn indicate deceptive subjects. Therefore, the features for a particular event (e.g., smiling) in each segment were first normalized by the total event count in that segment and the normalized by the duration (in seconds) for that segment, yielding normalized synchrony features, which you used to train classifiers to detect deceptive and truth-telling subjects.

Results of the automatic analysis and SVM classification are supportive of the initial hypothesis of the experiment. Table 5.4 shows classification accuracy (10-fold cross validation) using different classifiers trained on the selected $N = 31$ subjects. This means that monitoring synchrony events, while establishing implicit models of trust, can be useful for automatic deception detection.

Our analysis continues, as we are looking to see if the trend discovered thus far

by our computerized methods, generalizes to the greater sample population. However, we can safely conclude that normalization of synchrony event counts in each segment by the total event count in that segment is crucial to accurate deception recognition. Moreover, our synchrony attributes (i.e., event hits and event misses by each party) proved to be adequate quantitative measures of synchrony. What is even better is that deception detection through synchrony analysis achieves an accuracy close to the one achieved by our subject-specific modelling in the previous sections, with the difference that the method in this section does not require collection of training data to establish the baseline behavioral profiles of a new subject. Therefore, it has the potential to be a more practical method.

5.6 Summary

We presented two novel and fully automatic methods for deception detection from kinesic analysis on video input of interviews. One approach focused only on the subject and successfully attempted to distinguish over-controlled and agitated behaviors profiles from their respective baseline relaxed ones, using subject-specific modelling, outperforming state of the art methods. However, data from additional psychological studies of deception would help to further confirm that the behaviors discriminated by our learning algorithms are the deceptive behaviors we are attempting to isolate. The other approach looks at both the subject and their interviewer detecting accuracy via synchrony analysis and achieving similar accuracy. Both approaches have great potential and contributes to understanding deception detection from visual input in general, while the method relying on synchrony analysis has the added benefit that once trained it can be used without also requiring training data from the current test subject. Nevertheless, our results show a convincing proof of concept and suggest a promising future for the identification of deceptive behavior from video sequences.

Chapter 6

Conclusions and Future Research

Face tracking has numerous applications in the field of Human Computer Interaction and behavior understanding in general. Yet, face tracking is a difficult problem because the tracker must generalize to new faces, adapt to changing illumination, keep up with fast motions and pose changes, and tolerate target occlusion. Our objective in this work was to build an automatic face tracking system whose output can be used (directly or indirectly) to recognize dynamic events of facial expressions and head gestures, thus forming the foundation block of useful recognition applications. In fact, we focused on the application of our face tracking system to two very important recognition problems: (1) the recognition of non-manual markers (involving combinations of head gestures and facial expressions) in video sequences of American Sign Language, and (2) the automatic detection of deception in interview scenarios using kinesic analysis.

More specifically, we developed an extended a face tracker with stochastic shape transitions and a hierarchical observation likelihood models, resulting in better handling of head rotations and facial occlusions. We pioneered a framework for isolated recognition of non-manual markers in segmented ASL video sequences, using a "bag-of-words" model and later extended it to enable continuous recognition using sequential models and incorporating domain knowledge. We introduced a method for 2D image warping of non-frontal poses to frontal, using a trained 3D model to remove the effects of viewpoint changes on the appearance of an object and make the simplify the complexity of the recognition problem. Furthermore, we proved that deception detection is possible from kinesic analysis and from its more practical extension to synchrony analysis.

The methods presented in this thesis have applications to other domains as well. The

probabilistically extended face tracker can be applied to track any other deformable or rigid object, provided an appropriate training model. Similarly, the recognition methods we proposed for recognition of ASL markers can be used elsewhere to recognize combinations of head gestures and head movements by adjusting the feature set accordingly (since our framework is easily extensible), e.g, emotion recognition through analysis of facial expressions. This application can especially benefit from our face tracker, because it requires accurate facial tracking, and from our technique for image warping, which can help simplify the learning phase. More specifically, one may only need to train the various emotion recognition models using only labelled frontal image data, which is more widely available, and then apply our warping technique to every image in which the pose is non-frontal, before applying the trained recognition models. This will cut down the training time, because fewer models need to be trained and also less training data will be needed.

Moreover, there are a few improvements that can be made to our existing methods. Firstly, the face tracker can be made more specific by training additional models for the in-between poses (e.g., up right, up left, down right, down left, etc.) to better approximate the non-linear shape manifold. The combination factors for the observation likelihood model can be allowed to be dynamic, so as to weigh differently each of its components, based on the tracking context. A similar extension can be applied to the adaptive rate of the EOH spatial pyramid templates, as well as to the number of samples used by the particle filter. The sample set can be expanded as the cumulative density of the samples drops below a threshold to allow better sampling of the state space, and contracted while the cumulative density is high and the tracker is locked on target, so as to conserve computational resources. A further refinement would be to incorporate the mean-shift algorithm inside the particle filter for more efficient exploration of the state space by the sample set. This will improve accuracy but will also increase the running time per frame, which, however, can be decreased by efficient parallel code or even GPU programming, given that the samples can be forked, processed independently of each other and then merged again. Lastly, the 3D face model can be augmented with a texture model for more accurate image warping. The current technique begins to break

down, thankfully in a graceful manner, as the head rotation angle begins to increase past the 45^o limit. This is because at that point, certain areas of the face start getting occluded leading to discontinuities, so when the image is warped to frontal pose, there is not enough information in the visible area to interpolate how the occluded areas should be warped. However, if the 3D model is also trained with texture, then by matching the texture in the visible areas, the model can interpolate the texture of the occluded areas, too, for more fluid warping. This improvement is reminiscent of the 2D Active Appearance Models (AAM) but extended to 3D.

References

[1] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *International Conference on Machine Learning*, pages 3–10, 2003.

[2] O. Aran, T. Burger, A. Caplier, and L. Akarun. Gesture-based human-computer interaction and simulation. chapter Sequential Belief-Based Fusion of Manual and Non-manual Information for Recognizing Isolated Signs, pages 134–144. Springer-Verlag, Berlin, Heidelberg, 2009.

[3] B. Bauer and K.-F. Kraiss. Video-based sign recognition using self-organizing subunits. In *ICPR*, volume 2, pages 434–437, 2002.

[4] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In *NIPS*, 2003.

[5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[6] C. F. Bond and B. M. Depaulo. Accuracy of Deception Judgments. *Pers Soc Psychol Rev*, 10(3):214–234, Aug. 2006.

[7] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, pages 401–408, 2007.

[8] C. Bregler and S. Omohundro. Surface learning with applications to lipreading. 1994.

[9] P. Buddharaju, J. Dowdall, P. Tsiamyrtzis, D. Shastri, I. Pavlidis, and M. G. Frank. Automatic thermal monitoring system (ATHEMOS) for deception detection. *IEEE CVPR*, 2:1179, 2005.

[10] P. Buehler, A. Zisserman, and M. Everingham. Learning sign language by watching TV (using weakly aligned subtitles). In *IEEE CVPR*, pages 2961–2968, 2009.

[11] D. Buller, J. Burgoon, C. White, and A. Ebesu. Interpersonal deception: VII. Behavioral profiles of falsification, equivocation and concealment. *Journal of Language and Social Psychology*, 13:366–395, 1994.

[12] D. B. Buller, J. K. Burgoon, A. Buslig, and J. Roiger. Interpersonal deception: Viii. nonverbal and verbal correlates of equivocation from the bavelas et al. (1990) research. *Journal of Language and Social Psychology*, 13(4):396–417, 1994.

[13] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[14] J. K. Burgoon. A communication model of personal space violations: Explication and an initial test. *Human Communication Research*, 4:129–142, 1978.

[15] J. K. Burgoon. Nonverbal measurement of deceit. In V. Manusov, editor, *The sourcebook of nonverbal measures: Going beyond words*. Erlbaum, Hillsdale, NJ, 2005.

[16] J. K. Burgoon, M. L. Jensen, N. W. Twyman, T. O. Meservy, D. N. Metaxas, N. Michael, and J. F. Nunamaker. Automated kinesic analysis for deception detection. *Proceedings of the 43rd Hawaii International Conference on Systems Sciences (HICSS 2010)*, pages 31–40, 2010.

[17] P. Comon. Independent component analysis, a new concept? *Signal Process.*, 36:287–314, April 1994.

[18] H. Cooper and R. Bowden. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *IEEE CVPR*, pages 2568–2574, June 2009.

[19] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–573, 1999.

[20] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. In *Comp. Vis. Image Underst.*, pages 38–59, 1995.

[21] T. F. Cootes, K. Walker, and C. J. Taylor. View-based active appearance models. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 227–232, 2000.

[22] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 231–238, June 1996.

[23] D. DeCarlo and D. Metaxas. Deformable model-based shape and motion analysis from images using motion residual error. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, Washington, DC, USA, 1998. IEEE Computer Society.

[24] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *Int. J. Comput. Vision*, 38:99–127, July 2000.

[25] B. DePaulo, J. Lindsay, L. Muhlenbruck, K. Charlton, and H. Cooper. Cues to deception. *Psychological Bulletin*, 129:74–118, 2003.

[26] L. Ding and A. Martinez. Precise detailed detection of faces and facial features. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008.

[27] L. Ding and A. M. Martinez. Three-dimensional shape and motion reconstruction for the analysis of American Sign Language. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 146, Washington, DC, USA, 2006. IEEE Computer Society.

[28] D. F. Dinges, R. L. Rider, J. Dorrian, E. L. McGlinchey, N. L. Rogers, Z. Cizman, S. K. Goldenstein, C. Vogler, S. Venkataraman, and D. N. Metaxas. Optical computer recognition of facial expressions associated with stress induced by performance demands. *Aviation space and environmental medicine*, 76(6 Suppl):B172–B182, 2005.

[29] DoDPI R. D. Staff, J. L. Meyerhoff, G. A. Saviolakis, M. L. Koening, and D. L. Yourick. Physiological and biochemical measures of stress compared to voice stress analysis using the computer voice stress analyzer (CVSA) (Report No. DoDPI98-R-0004). Department of Defense Polygraph Institute, Ft. Jackson, SC. 2001.

[30] G. Edwards, C. Taylor, and T. Cootes. Learning to identify and track faces in image sequences. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 260–265, April 1998.

[31] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, FG '98, Washington, DC, USA, 1998. IEEE Computer Society.

[32] P. Ekman. Facial signs: Facts, fantasies, and possibilities. *Sight sound and sense*, pages 124–156, 1978.

[33] P. Ekman. Lying and nonverbal behavior: Theoretical issues and new findings. *Journal of Nonverbal Behavior*, 12:163–176, 1988.

[34] P. Ekman. *Telling lies: Clues to deceit in the marketplace, politics, and marriage*, volume 2. WW Norton and Company, New York, 1992.

[35] P. Ekman and W. V. Friesen. Nonverbal leakage and clues to deception. *Psychiatry*, 32:88–106, 1969.

[36] P. Ekman, W. V. Friesen, and J. C. Hager. *The Facial Action Coding System*. Research Nexus eBook, Salt Lake City, UT, USA, 2002.

[37] P. Ekman and M. O'Sullivan. Who Can Catch a Liar? *American Psychologist*, 46(9):913–920, 1991.

[38] U. Erdem and S. Sclaroff. Automatic detection of relevant head gestures in american sign language communication. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 460–463, 2002.

[39] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vision*, 61:55–79, January 2005.

[40] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. *AFGRW*, pages 296–301, 1995.

[41] G. Ganis, S. M. Kosslyn, S. Stose, W. L. Thompson, and D. A. Yurgelun-Todd. Neural correlates of different types of deception: An fmri investigation. *Cerebral Cortex*, 13:830–836, 2003.

[42] J. C. Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders. Kernel codebooks for scene categorization. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 696–709, Berlin, Heidelberg, 2008. Springer-Verlag.

[43] J. George, D. P. Biros, J. K. Burgoon, and J. Nunamaker. Training professionals to detect deception. *NSF/NIJ Symposium on Intelligence and Security Informatics*, Tucson, AZ, 2003.

[44] S. Goldenstein, C. Vogler, and D. Metaxas. 3d facial tracking from corrupted movie sequences. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–880 – I–885, 2004.

[45] S. Gong, E.-J. Ong, and S. J. McKenna. Learning to associate faces across views in vector space of similarities to prototypes. In *BMVC*, 1998.

[46] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(2):285–339, 1991.

[47] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, October 2005.

[48] L. Gu and T. Kanade. A generative shape regularization model for robust face alignment. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 413–426, Berlin, Heidelberg, 2008. Springer-Verlag.

[49] T. Heap and D. Hogg. Improving specificity in pdms using a hierarchical approach. In A. F. Clark, editor, *BMVC*. British Machine Vision Association, 1997.

[50] T. Heap and D. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, Washington, DC, USA, 1998. IEEE Computer Society.

[51] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[52] F. Jiao, S. Li, H.-Y. Shum, and D. Schuurmans. Face alignment using statistical models and wavelet features. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–321 – I–327, June 2003.

[53] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77:27–59, 2009.

[54] R. Johnson, J. Barnhardt, and J. Zhu. The contribution of executive processes to deceptive responding. *Neuropsychologia*, 42:878–901, 2004.

[55] M. Jones and J. Rehg. Statistical color models with application to skin detection. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 274–280, 1999.

[56] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[57] A. Kanaujia, Y. Huang, and D. Metaxas. Tracking facial features using mixture of point distribution models. In *ICVGIP*, 2006.

[58] S. Kassin, R. Leo, C. Meissner, K. Richman, L. Colwell, A.-M. Leach, and D. Fon. Police interviewing and interrogation: A self-report survey of police practices and beliefs. *Law and Human Behavior*, 31:381–400, 2007.

[59] D. Kelly, J. Delannoy, J. McDonald, and C. Markham. Incorporating facial features into a multi-channel gesture recognition system for the interpretation of irish sign language sequences. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1977–1984, 2009.

[60] D. Kelly, J. Reilly Delannoy, J. Mc Donald, and C. Markham. A framework for continuous multimodal sign language recognition. In *Proceedings of the 2009 international conference on Multimodal interfaces*, ICMI-MLMI '09, pages 351–358, New York, NY, USA, 2009. ACM.

[61] D. Kelly, J. Reilly Delannoy, J. McDonald, and C. Markham. Automatic recognition of head movement gestures in sign language sentences. In *Proceedings of the China-Ireland Information and Communications Technologies Conference (CIICT)*, pages 142–145, 2009.

[62] S. Klein Goldenstein, C. Vogler, and D. Metaxas. Statistical cue integration in dag deformable models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(7):801–813, July 2003.

[63] F. A. Kozel, K. A. Johnson, Q. Mu, E. L. Grenesko, S. J. Laken, and M. S. George. Detecting deception using functional magnetic resonance imaging. *Biological Psychiatry*, 58(8):605–613, 2005.

[64] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE CVPR*, pages 2169–2178, 2006.

[65] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43:29–44, 2001.

[66] Y. Li and W. Ito. Shape parameter optimization for adaboosted active shape model. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 251–258, October 2005.

[67] S. K. Liddell. *American Sign Language Syntax*. Mouton, The Hague, 1980.

[68] Z. Lin, G. Hua, and L. Davis. Multiple instance feature for robust part-based object detection. *IEEE Conf. on Computer Vision and Pattern Recognition*, 0:405–412, 2009.

[69] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[70] S. Lu, G. Tsechpenakis, D. Metaxas, M. L. Jensen, and J. Kruse. Blob analysis of the head and hands: A method for deception detection and emotional state identification. In *Hawaii International Conference on System Sciences*, Big Island, Hawaii, January 2005.

[71] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems*, volume 10, pages 570–576. MIT Press, 1998.

[72] T. Meservy, M. Jensen, J. Kruse, J. Burgoon, J. Nunamaker, J.F., D. Twitchell, G. Tsechpenakis, and D. Metaxas. Deception detection through automatic, unobtrusive analysis of nonverbal behavior. *Intelligent Systems, IEEE*, 20(5):36 – 43, 2005.

[73] T. O. Meservy, M. L. Jensen, J. Kruse, J. K. Burgoon, and J. F. Nunamaker. *Automatic Extraction of Deceptive Behavioral Cues from Video*, volume 3495/2005 of *LNCS*, pages 198–208. Springer Berlin/Heidelberg, 2005.

[74] N. Michael, M. Dilsizian, D. Metaxas, and J. K. Burgoon. Motion profiles for deception detection using visual cues. *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, 2010.

[75] N. Michael, D. N. Metaxas, and C. Neidle. Spatial and temporal pyramids for grammatical expression recognition of American Sign Language. In *ASSETS*, pages 75–82, October 2009.

[76] N. Michael, C. Neidle, and D. Metaxas. Computer-based recognition of facial expressions in ASL: from face tracking to linguistic interpretation. *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (LREC)*, May 2010.

[77] N. Michael, P. Yang, Q. Liu, D. Metaxas, and C. Neidle. A framework for the recognition of non-manual markers in segmented sequences of American Sign Language. *Proceedings of the 22nd British Machine Vision Conference (BMVC)*, August 2011.

[78] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *Proceedings of the 10th European Conference on Computer Vision: Part IV*, pages 504–513, Berlin, Heidelberg, 2008. Springer-Verlag.

[79] K. W. Ming and S. Ranganath. Representations for facial expressions. In *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, volume 2, pages 716–721, December 2002.

[80] H. Murase and S. Nayar. Learning and recognition of 3d objects from appearance. In *Qualitative Vision, 1993., Proceedings of IEEE Workshop on*, pages 39–50, June 1993.

[81] J. Navarro. *What every BODY is saying: an ex-FBI agent's guide to speed-reading people*. Collins Living, New York, NY, 2008.

[82] C. Neidle. Signstream$^{\text{TM}}$: A Database Tool for Research on Visual-Gestural Language, 2000. Boston MA: American Sign Language Linguistic Research Project No. 10, Boston University.

[83] C. Neidle, J. Kegl, D. MacLaughlin, B. Bahan, and R. G. Lee. *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure*. MIT Press, Cambridge MA, 2000.

[84] C. Neidle, N. Michael, J. Nash, and D. Metaxas. A method for recognition of grammatically significant head movements and facial expressions, developed through use of a linguistically annotated video corpus. *Proc. of 21st ESSLLI Workshop on Formal Approaches to Sign Languages*, July 2009.

[85] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 14:849–856, 2002.

[86] T. D. Nguyen and S. Ranganath. Towards recognition of facial expressions in sign language: Tracking facial features under occlusion. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 3228–3231, October 2008.

[87] T. D. Nguyen and S. Ranganath. Tracking facial features under occlusions and recognizing facial expressions in sign language. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–7, September 2008.

[88] T. D. Nguyen and S. Ranganath. Recognizing continuous grammatical marker facial gestures in sign language video. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part IV*, ACCV'10, pages 665–676, Berlin, Heidelberg, 2011. Springer-Verlag.

[89] T. Ojala and M. Pietikäinen. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 2002.

[90] M. A. Oleshansky and J. L. Meyerhoff. Acute catecholaminergic responses to mental and physical stressors in man. *Stress Medicine*, 8:175–179, 1992.

[91] S. C. W. Ong and S. Ranganath. Automatic sign language analysis: a survey and the future beyond lexical meaning. *IEEE TPAMI*, 27(6):873–891, June 2005.

[92] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE TPAMI*, 19:677–695, 1997.

[93] F. Porikli. Integral histogram: a fast way to extract histograms in cartesian spaces. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 829–836, June 2005.

[94] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[95] M. Rogers and J. Graham. Robust active shape model search. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, ECCV '02, pages 517–530, London, UK, UK, 2002. Springer-Verlag.

[96] S. Romdhani, S. Gong, and A. Psarrou. A multi-view nonlinear Active Shape Model using kernel PCA. *British machine vision conference*, pages 483–492, 1999.

[97] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *Int. J. Comput. Vision*, 91:200–215, January 2011.

[98] S. Sarkar, B. Loeding, and A. Parashar. Fusion of manual and non-manual information in american sign language recognition. In C. H. Chen, editor, *Handbook of Pattern Recognition and Computer Vision (4th Edition)*. CRC Press, 2010.

[99] C. Shan, S. Gong, and P. W.McOwan. Robust facial expression recognition using local binary patterns. *IEEE Int. Conf. on Image Processing*, 2005.

[100] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, June 1994.

[101] L. Sigal, S. Sclaroff, and V. Athitsos. Skin color-based video segmentation under time-varying illumination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(7):862–877, July 2004.

[102] T. Starner and A. Pentland. Visual recognition of American Sign Language using Hidden Markov Models. In *International Workshop on Automatic Face and Gesture Recognition*, pages 189–194, 1995.

[103] C. Sutton and A. McCallum. Introduction to statistical relational learning. chapter An introduction to conditional random fields for relational learning. MIT Press, 2006.

[104] M. J. Swain and D. H. Ballard. Color indexing. *IJCV*, 7:11–32, 1991.

[105] X. Tan, F. Song, Z.-H. Zhou, and S. Chen. Enhanced pictorial structures for precise eye localization under incontrolled conditions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1621–1628, June 2009.

[106] Y. Tian. Evaluation of face resolution for expression analysis. *Computer Vision and Pattern Recognition Workshop on Face Processing in Video*, 2004.

[107] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[108] C.-F. Tsai and C. Hung. Automatically annotating images with keywords: A review of image annotation systems. *Recent Patents on Computer Science*, 1:55–68, 2008.

[109] G. Tsechpenakis, D. Metaxas, M. Adkins, J. Kruse, J. Burgoon, M. Jensen, T. Meservy, D. Twitchell, A. Deokar, and J. Nunamaker. HMM-based deception recognition from visual cues. In *IEEE ICME*, pages 824–827, Los Alamitos, CA, USA, 2005. IEEE.

[110] P. Tsiamyrtzis, J. Dowdall, D. Shastri, I. T. Pavlidis, M. G. Frank, and P. Ekman. Imaging facial physiology for the detection of deceit. *International Journal of Computer Vision*, 71:197–214, February 2007.

[111] B. Turvey. United States v. Gordon E. Thomas, III - a case study in the reliability of criminal investigative analysis. *Journal of Behavioral Profiling*, 7(1), 2007.

[112] M. Valstar and M. Pantic. Combined Support Vector Machines and Hidden Markov Models for modeling facial action temporal dynamics. In *HCI'07: Proceedings of the IEEE Workshop on Human Computer Interaction*, pages 118–127, 2007.

[113] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[114] C. Vogler and S. Goldenstein. Facial movement analysis in ASL. *Univers. Access Inf. Soc.*, 6(4):363–374, 2008.

[115] C. Vogler and S. Goldenstein. Toward computational understanding of sign language. *Technology and Disability*, 20(2):109–119, 2008.

[116] C. Vogler, S. Goldenstein, J. Stolfi, V. Pavlovic, and D. Metaxas. Outlier rejection in high-dimensional deformable models. *Image Vision Comput.*, 25:274–284, March 2007.

[117] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *ICCV*, pages 363–369, 1998.

[118] C. Vogler and D. Metaxas. *Handshapes and movements: Multiple-channel ASL recognition*, pages 247–258. LNAI. Springer, Berlin, 2004.

[119] U. von Agris, M. Knorr, and K.-F. Kraiss. The significance of facial features for automatic sign language recognition. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–6, September 2008.

[120] U. von Agris, D. Schneider, J. Zieren, and K.-F. Kraiss. Rapid signer adaptation for isolated sign language recognition. In *V4HCI*, 2006.

[121] U. von Agris, J. Zieren, U. Canzler, B. Bauer, and K.-F. Kraiss. Recent developments in visual sign language recognition. *Univers. Access Inf. Soc.*, 6(4):323–362, 2008.

[122] A. Vrij. *Detecting lies and deceit: The psychology of lying and its implications for professional practice*. Wiley, Chichester, UK, 2000.

[123] A. Vrij, K. Edward, K. Roberts, and R. Bull. Detecting deceit via analysis of verbal and nonverbal behavior. *Journal of Nonverbal Behavior*, 24:239–263, 2000.

[124] Y. Wang, X. Ni, and F. Jian. Discriminative training methods of HMM for sign language recognition. *CAAI Transactions on Intelligent Systems*, 1:80–84, 2007.

[125] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[126] M. Xu, 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. In *Proceedings of the Third International Conference on Advances in Multimodal Interfaces*, ICMI '00, pages 572–581, London, UK, 2000. Springer-Verlag.

[127] F. Yang, J. Huang, and D. Metaxas. Sparse shape registration for occluded facial feature localization. In *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 272–277, March 2011.

[128] F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3d-aware face component transfer. 2011.

[129] H.-D. Yang and S.-W. Lee. Combination of manual and non-manual features for sign language recognition based on conditional random field and active appearance model. In *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, volume 4, july 2011.

[130] P. Yang, Q. Liu, and D. N. Metaxas. Boosting encoded dynamic features for facial expression recognition. *Pattern Recognition Letters*, 30(2):132–139, 2009.

[131] Z. Zhang, V. Singh, T. E. Slowe, S. Tulyakov, and V. Govindaraju. Real-time automatic deceit detection from involuntary facial expressions. In *IEEE CVPR*, 2007.

[132] G. Zhao and M. Pietikäinen. Dynamic texture recognition using volume local binary patterns. *European Conference on Computer Vision*, 2006.

[133] Y. Zhou, L. Gu, and H.-J. Zhang. Bayesian tangent shape model: estimating shape and pose parameters via bayesian inference. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–109 – I–116, June 2003.

[134] Y. Zhou, W. Zhang, X. Tang, and H. Shum. A bayesian mixture model for multi-view face alignment. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 741–746, June 2005.

[135] Q. Zhu, K.-T. Cheng, C.-T. Wu, and Y.-L. Wu. Adaptive learning of an accurate skin-color model. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 37 – 42, May 2004.

[136] J. Zieren and K.-F. Kraiss. Robust person-independent visual sign language recognition. In *Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis*, volume LNCS, 2005.

[137] M. Zuckerman, B. M. DePaulo, and R. Rosenthal. Verbal and nonverbal communication of deception. *Advances in Experimental Social Psychology*, 14:1–59, 1981.

Chapter 7
Curriculum Vitae

Education

- **Rutgers University**, New Brunswick, NJ — 09/2006 - 01/2012
  Ph.D. candidate in Computer Science (GPA 3.88/4.00)
- **Rutgers University**, New Brunswick, NJ — 09/2006 - 05/2011
  M.Phil., in Computer Science (GPA 3.88/4.00)
- **Rutgers University**, New Brunswick, NJ — 09/2006 - 05/2011
  M.S., in Computer Science (GPA 3.88/4.00)
- **SUNY Binghamton University**, Binghamton, NY — 09/2002 - 05/2006
  B.S., Computer Science (GPA 3.91/4.00)
- **SUNY Binghamton University**, Binghamton, NY — 09/2002 - 05/2006
  B.A., Mathematics – Actuarial track (GPA 3.91/4.00)

Experience

- **Research Assistant**, Rutgers University — 07/2008 - 12/2011
- **Research Assistant**, University of Cyprus — 05/2007 - 07/2007
- **Teaching Assistant**, Rutgers University — 09/2006 - 06/2008
- **Military Service**, National Guard, Cyprus — 07/2000 - 08/2002

Publications

1. **[BMVC'11]** N. Michael, P. Yang, Q. Liu, D. Metaxas, C. Neidle: A framework for the recognition of non-manual markers in segmented sequences of American Sign Language, *22nd British Machine Vision Conference*, 2011.

2. **[IAA'11a]** N. Michael, F. Yang, D. Metaxas and D. Dinges: Development of optical computer recognition (OCR) for monitoring stress and emotions in space, 18th IAA Humans in Space Symposium, 2011.

3. **[IAA'11b]** F. Yang, N. Michael, D. Metaxas and D. Dinges, "Development of optical computer recognition (OCR) for monitoring fatigue in space", 18th IAA Humans in Space Symposium, 2011.

4. **[ECCV'10]** N. Michael, M. Dilsizian, D. Metaxas, J. K. Burgoon: Motion Profiles for Deception Detection using Visual Cues, 11th European Conference on Computer Vision, 2010.

5. [**HICSS'10**] J. K. Burgoon, M. L. Jensen, N. W. Twyman, T. O. Meservy, D. N. Metaxas, N. Michael, J. F. Nunamaker, Jr.: Automated Kinesic Analysis for Deception Detection, 43^{rd} Hawaii International Conference on Systems Sciences (HICSS) – Proceedings of the Credibility Assessment and Information Quality in Government and Business Symposium, 2010.

6. [**LREC'10**] N. Michael, C. Neidle, and D. Metaxas: Computer-based recognition of facial expressions in ASL: from face tracking to linguistic interpretation, 4^{th} Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (LREC), 2010.

7. [**CDDA'10**] N. Michael, F. Yang, M. Dilsizian and D. Metaxas: Facial Tracking for Behavior and Gesture Recognition, Center for Dynamic Data Analytics Workshop, 2010.

8. [**ASSETS'09**] N. Michael, D. Metaxas, and C. Neidle: Spatial and Temporal Pyramids for Grammatical Expression Recognition of American Sign Language, 11^{th} International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS), 2009.

9. [**ESSLI'09**] C. Neidle, N. Michael, J. Nash, and D. Metaxas: A Method for Recognition of Grammatically Significant Head Movements and Facial Expressions, Developed through use of a Linguistically Annotated Video Corpus, Workshop on Formal Approaches to Sign Languages, 2009.