Automatic Facial Action Analysis

by

Ashish Kapoor

Bachelor of Technology in Computer Science & Engineering
Indian Institute of Technology, Delhi
August 2000

Submitted to the Program in Media Arts & Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author .....
Program in Media Arts & Sciences
May 10, 2002

Certified by .....
Rosalind W. Picard
Associate Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by .....
Andrew B. Lippman
Chairperson
Departmental Committee on Graduate Students

Automatic Facial Action Analysis

by

Ashish Kapoor

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on May 10, 2002, in partial fulfillment of the
requirements for the degree of
MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES

Abstract

This thesis provides a fully automatic framework to analyze the facial actions and head gestures in real time. This framework can be used in scenarios where the machine needs a perceptual ability to recognize, model and analyze the facial actions and head gestures in real time without any manual intervention. Rather than trying to recognize specific prototypical emotional expressions like joy, anger, surprise and fear, this system aims to recognize the head gestures and the upper facial action units such as eyebrow raises, frowns and squints. These facial action units (AUs) are enumerated in Paul Ekman's Facial Action Coding System (FACS) [17] and are essentially building blocks, which can be assembled to form facial expressions. The system first robustly tracks the pupils using an infrared sensitive camera equipped with infrared LEDs. For each frame, the pupil positions are used to localize regions of eyes and eyebrow, which are analyzed using statistical techniques to recover parameters that relate to the shape of the facial features. These parameters are used as input to classifiers based on Support Vector Machines to recognize upper facial action units and their all possible combinations. The system detects head gestures using Hidden Markov Models that use pupil positions in consecutive frames as observations. The system is evaluated on completely natural dataset with lots of head movements, pose changes and occlusions. The system can successfully detect head gestures 78.46% of time. Recognition accuracy of 67.83% for each individual AU is reported and the system can correctly identify all possible AU combinations with an accuracy of 61.25%.

Thesis Supervisor: Rosalind W. Picard,
Associate Professor of Media Arts and Sciences.

This research was supported by NSF ROLE grant 0087768 and also is an output from a research project funded by Media Lab Asia. Media Lab Asia is funded in part by the Ministry of Information Technology, Government of India. The research was carried out in support of the Media Lab Asia Program by Massachusetts Institute of Technology's Media Laboratory itself. Media Lab Asia does not accept responsibility for any information provided or views expressed.

Automatic Facial Action Analysis

by

Ashish Kapoor

Thesis Reader .....

Alex (Sandy) Pentland

Toshiba Professor of Media Arts and Sciences

Massachusetts Institute of Technology

Thesis Reader .....

Trevor Darrell

Assistant Professor of Electrical Engineering and Computer Science

Massachusetts Institute of Technology

Acknowledgments

I would like to thank Roz Picard for bringing me to MIT in the first place and having so much faith in me. Thank you Roz, for being so patient and supportive, and for motivating me with your inspirational ideas.

I thank Trevor Darrell and Sandy Pentland for being readers for this thesis. Thanks to Sandy for being a great source of inspiration and enthusiasm. Thanks to Trevor for his great ideas and insight.

I wish to thank all the folks who helped with this thesis. Special thanks to Nancy Alvarado, who tirelessly coded the videos for me. Thanks to Stefan Agamanolis for his help with the video coding. Thanks to Rob Reilly and Barry Kort for their wisdom and experience with the kids. I would also like to thank Justine Cassell who taught me that there is more to the world than just the computers. Thanks to my UROPs, Yue Hann Chin, David Lopez Mateos and Tim Heidel for helping me. Thanks to Raul Fernandez, Yuan Qi, Carson Reynolds and Stoffel Kuenen for they insight and invaluable council. And special thanks to Selene Mota for being such a wonderful office mate and a friend. Thanks to the Pentlandians, Sumit Basu, Tanzeem Choudhury, Brian Clarkson, Ali Rahimi, Tony Jebara, Vikram Sheel Kumar, Vishwanath Anantraman and Yuri Ivanov for their support and inspiration.

Thanks to all my friends who stood by me all the time. Thanks to Erin Maneri, Kate Lesnaia, Julian Lange, Sanjay Jain, Ashish Nimgaonkar and Vipin Gupta. Thanks to Anshul Sood and Anurag Chandra for helping me settle down here and being there when I needed them the most.

Thanks to my family and friends back from home and finally, thanks to Mom, Dad and my sister Aakashi for their unconditional love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A very large percentage of our communication is nonverbal and among these nonverbal cues a large fraction is in the form of facial actions. The facial actions perform a number of different functions. Besides telling us about the affective and cognitive state of a person [20], they are used as social and conversational cues and perform semantic functions as well [7]. A system that could analyze the facial actions in real time without any human intervention will have applications in a number of different fields: for example, computer vision, affective computing, computer graphics and psychology. Such a system will be an important component in a machine that is socially and emotionally intelligent and is expected to interact naturally with people.

The problem of automatic face analysis is a hard one. The face is an immense source of information about the psychological, physiological and cognitive state of a person. This information can be thought of as signals/observations that are emitted by the underlying hidden state of the person. These facial signals are very complicated and we need a system to quantify these observations. A lot of people have used emotional expressions like happy/sad/angry to quantify these facial signals. Although this kind of quantification scheme might suggest a direct relationship between underlying emotions and facial expressions, the social, cultural and circumstantial variables make this relationship very complicated. So, rather then using high level descriptions like happiness, anger etc., we need a more basic quantifying scheme. The Facial Action Coding System (FACS) developed by Ekman and Friesen [17] is

Table 1.1: Comparison of various face analysis systems

	Real time	**Fully automatic**	**Recognize more than prototype expressions**
Black & Yacoob [3] 1995	No	No	No
Esaa et al [18] 1997	Yes	Yes	No
Tian et al [31] 2000	No	No	Yes

a method of measuring facial activity in terms of facial muscle movements. FACS consists of over 45 distinct action units corresponding to a distinct muscle or muscle group. Though FACS has been criticized as only capturing a spatial description of facial activity and ignoring the temporal component, it is perhaps the most widely used language to describe facial activity at the muscle level. It is a standard system which has been used in behavioral sciences for years.

While a lot of research has been directed towards systems that recognize faces corresponding to prototypic expressions like joy and surprise, few approaches exist that try to recognize facial actions such as eye-squint and frown. Table 1.1 compares some of the previous facial expression analysis techniques. The state of the art systems have severe limitations as they either require human intervention or do not recognize more than prototypic expressions.

1.1 Thesis Objective

The purpose of this thesis is to develop a fully automatic framework that requires no manual intervention to analyze facial activity in real time. The work is focused on recognizing head gestures and upper AUs. These upper AUs are a subset of all the AUs enumerated in Paul Ekman's Facial Action Coding System (FACS) [17] and correspond to the regions of eyes and eyebrows (Table 1.2). In addition, the thesis

Table 1.2: The upper facial action units

AU number	Facial action
1	Inner brow raiser
2	Outer brow raiser
4	Brow lowerer
5	Upper eye lid raiser
6	Cheek raiser
7	Lid tightener

addresses the issues of how to use statistical learning methods to automatically recover parameters describing facial features and model them to analyze the facial actions and recognize head gestures.

1.2 Application Scenarios in Computer Human Interaction

A lot of research is being done to make machines socially and emotionally intelligent. The machine that could understand behavioral cues about various emotional and social situations can influence how humans interact with machines. This kind of system can be used to assist humans in tasks that require people to make decisions based on a number of social and emotional variables. One such system is the learning companion, which is an affective peer/teacher that helps students through their learning journey. A critical component of the learning companion is an affect recognition system and the face analysis system is a part of this component. The work in this thesis draws its motivation from the learning companion and the data used for training and testing of the facial actions was gathered in the learning companion scenario. Below are the details of the learning companion and the data used for the purpose of facial action recognition. Although the work is focused on recognizing facial actions in a learning situation, the ideas extend to other application scenarios as well.

Table 1.3: Surface level behaviors

	On-Task	Off-Task
Facial Actions	Eyes tightening (AU 7), widening (AU 5), Raising eyebrows (AU 1+2), Smile (AU 6+12)	Lowering eyebrow (AU 1+4), Nose wrinkling Depressing lower lip corner (AU 15)
Posture	Leaning forward, Sitting upright	Slumping on the chair, Fidgeting
Eye-Gaze	Looking towards the problem	Looking everywhere else
Head Nod/ Head Shake	Up-down head nod	Sideways head shake
Hand Movement	Typing, clicking mouse	Hands not on mouse/keyboard

1.2.1 Learning Companion: An Affective Tutor

Learning the complex ideas involved in science, math, engineering, and technology and developing the cognitive reasoning skills that these areas demand often involves failure and a host of associated affective responses. These affective responses can range from feelings of interest and excitement to feelings of confusion and frustration. The student might quit if he is not able to recover from the 'feeling of getting stuck'. Expert teachers are very adept at recognizing and addressing the emotional state of learners and based upon that observation taking some action that positively impacts learning. One of the aims of the learning companion project at the MIT Media lab is to build a computerized learning companion that can do that.

Skilled humans can assess emotional signals with varying degrees of accuracy, and researchers are beginning to make progress giving computers similar abilities at recognizing affective expressions. Computer assessments of a learner's emotional state can be used to influence how and when an automated companion chooses to intervene.

The Learning Companion aims to sense emotional and cognitive aspects of the

learning experience in an unobtrusive way. Cues like posture, gesture, eye gaze and facial expression help expert teachers to recognize whether the learner is on-task or off-task. Affective states in learning (like interest/ boredom/ confusion/ excitement) are accompanied by different patterns of postures, gesture, eye-gaze and facial expressions. These surface level behaviors and their mappings are loosely summarized in Table 1.3. Whether all of these are important, and are the right ones remains to be evaluated, and it will no doubt take many investigations. Such a set of behaviors may be culturally different and will likely vary with developmental age as well. The point is that there are variety of surface level behaviors related to inferring the affective state of the user, while he or she is engaged in natural learning situations.

This thesis is primarily focused on analyzing faces. The facial expressions and head gestures are good indicators of affective and motivational states. Reeves [27] showed that the number of eye glances, the duration of eye glances, the number of times eyes were closed, smiles, head turns and head stillness correlate with self-reported interest for subjects watching movies. The interaction in learning companion is very different from the one that was used in the study and we believe that there are more facial actions correlated with the emotional states important to the learning companion. Approving head nods and facial actions like smile (AU 6+12), tightening of eyelids while concentrating (AU 7), eyes widening (AU 5) and raising of eyebrows (AU 1+2) might suggest interest/ surprise/ excitement (on task), whereas head shakes, lowering of eyebrows (AU 1+4), nose wrinkling (AU 9) and depressing lower lip corner (AU 15) might suggest the state off-task. This work is focused on detecting the above mentioned AUs (except AU 9,12 and 15) and the training and testing data for the purpose of facial action recognition was collected in a real learning scenario, as described below.

1.2.2 Data Collection and Annotation

25 kids ranging from 8 years to 11 years were invited to participate in an experiment. These kids were asked to solve a number of puzzles that required mathematical reasoning. Videos of their faces were recorded by two cameras. A vision camera was

placed on top of the monitor and an IBM Blue Eyes camera was placed under the monitor. The IBM Blue Eyes camera [25] is an infrared camera equipped with infrared LEDs that helps in pupil tracking (see chapter 3). A FACS trained expert coded the videos of the face for various action units. The video shot through the IBM Blues Eyes camera was used as a source of both training and testing data for facial action recognition.

1.3 Outline of the Thesis

This thesis focuses on two issues. First how to extract parameters that would describe the facial features and second how to use these parameters to recognize facial actions and head gestures. To answer these questions, I explore methods in statistical learning to develop groundwork and experiment with this real world data to establish its usefulness. The organization of this thesis is as follows:

- Chapter 2 reviews the prior work related to face analysis and related research in the areas of machine learning and computer vision.
- Chapter 3 gives the overview of the system with emphasis on pupil tracking.
- Chapter 4 addresses example based learning to recover the shape parameters of facial features in real time.
- Chapter 5 concerns real-time recognition of facial actions and head gestures using statistical machine learning techniques.
- Chapter 6 summarizes the result, discusses potential applications and concludes the thesis with suggestions for future work.

Chapter 2

Related Work

There has been a lot of research on building systems to automatically analyze the face. Most of this research has focused on systems that use computer vision and pattern recognition techniques to passively sense the facial activity. Most of these systems can be broadly classified into two categories:

- Systems that recognize prototypic facial expressions corresponding to basic emotions. For example: happy/sad/angry etc.
- Systems that recognize facial actions. For example: frown, eyebrow raise, nose wrinkle etc.

This chapter gives a brief overview of some of these approaches. Both kinds of approaches need to extract features or meaningful information from the image sequences for the purpose of recognition. These features can be optical flows, Gabor wavelet representations, geometrical features or any set of parameters that could model the facial activity. This chapter also describes related work in facial feature extraction. As it is impossible to cite all the work on face analysis, the following are some of the important ideas that lay the foundation for this thesis.

In passing, I would like to mention that there have been approaches that try to sense the facial movement using wearable sensors like masks or glasses. For example, Expression Glasses built by Scheirer et al [28] can sense upward eyebrow activity indicative of expressions such as interest and downward eyebrow activity indicative

of confusion or dissatisfaction. These wearables are more accurate in sensing the facial activity than a camera, but they are more physically intrusive.

2.1 Recognizing Facial Expressions

Ekman and Friesen [15] [16] have proposed that there are some prototypic facial expressions that are universal and correspond to basic human emotions. Based on this, a lot of research has been directed at the problem of recognizing 5-7 classes of prototypic emotional expressions on groups of people from their facial expressions.

Black and Yacoob [3] describe a system that recognizes facial expressions in presence of significant head motions. They use parameterized optical flow models to track rigid and non-rigid facial movements. In an earlier version Yacoob and Davis [34] use optical flow at high gradient points on the face to recognize facial expressions. Essa and Pentland [18] analyze the facial expressions using optical flow in an estimation and control framework coupled with a physical model describing the skin and muscle structure of face. Zhang [37] has compared the use of geometrical features with a multi-scale, multi-orientation Gabor wavelet based representation to identify expressions.

Although prototypic expressions, like happy, surprise and fear, are natural, they occur infrequently in everyday life. A person might communicate more with subtle facial actions like frequent frowns or smiles. Further there are emotions like confusion, boredom and frustration for which any prototypic expression might not exist. Thus, a system that aims to be socially and emotionally intelligent needs to do more than just recognize prototypic expressions.

2.2 Recognizing Facial Actions

Very little facial expression analysis research has focused on recognizing specific facial actions like raising an eyebrow, squinting and depressing the lip corners. Choudhury [6] has demonstrated a system to recognize some basic facial actions like mouth ac-

tivity and eye-movements using Hidden Markov Models (HMMs) and multidimensional receptive field histograms. Her system only recognizes basic facial actions like blinks/mouth open/eyes closed etc. and cannot recognize more subtle facial actions like eye-widening.

Kawato and Ohya [23] have described a system to detect head nods and head shakes in real time by directly detecting and tracking the "between-eyes" region. The "between-eyes" region is detected and tracked using a "circle frequency filter", which is a discrete Fourier transform of points lying on a circle, together with skin color information and templates. Head nods and head shakes are detected based on pre-defined rules applied to the positions of "between-eyes" in consecutive frames.

Motivated by Paul Ekman's Facial Action Coding System (FACS), some of the approaches attempt to recognize action units (AUs) - the fundamental muscle movements that comprise Paul Ekman's Facial Action Coding System, which can be combined to describe all facial expressions [17]. These facial actions are essentially facial phonemes, which can be assembled to form facial expressions.

Tian et al [31] have developed a system to recognize sixteen action units and any combination of those. The shape of facial features like eyes, eyebrow, mouth and cheeks are described by multistate templates. The parameters of these multistate templates are used by a Neural Network based classifier to recognize the action units. This system requires that the templates be initialized manually in the first frame of the sequence, which prevents it from being fully automatic. In an earlier work, Lien et al [24] describe a system that recognizes various action units based on dense flow, feature point tracking and edge extraction.

Donato et al [13] compared several techniques, which included optical flow, principal component analysis, independent component analysis, local feature analysis and Gabor wavelet representation, to recognize eight single action units and four action unit combinations using image sequences that were manually aligned and free of head motions. They showed 95.5% recognition accuracy using Independent Component Analysis and Gabor wavelet representations. Bartlett et al [1] achieve 90.9% accuracy in recognizing 6 single action units by combining holistic facial analysis and

optical flow with local feature analysis. Both of the above approaches report their results on manually pre-processed image sequences of individuals deliberately making facial actions in front of a camera.

Cowie et al [12] describe a system to recognize facial expressions by identifying *Facial Animation Parameter Units (FAPUs)* defined in MPEG-4 standard by feature tracking of *Facial Definition Parameter(FDP)* points, also defined in MPEG-4 framework. The system is not fully automatic and requires human assistance to accurately detect FDP points.

2.3 Facial Feature Tracking

Facial action analysis requires extraction of features, either physical (features like eyes, brows etc) or appearance based (Optical flows, Gabor coefficients etc that represent movements and positions of facial feature).

There is much prior work on detecting and tracking facial features. Many feature extraction methods are based on deformable templates [36], which are difficult to use for real-time tracking and have to be initialized properly to achieve a good performance. Tian et al [29, 30] use multiple-state templates to track the facial features. Feature point tracking together with masked edge filtering is used to track the upper facial features. The system requires that templates be manually initialized in the first frame of the sequence, which prevents it from being automatic.

There have been other approaches that do not depend upon computer vision techniques only. Morimoto et al [25] have described a system to detect and track pupils using the red-eye effect. Haro et al [21] have extended this system to detect and track the pupils using a Kalman filter and probabilistic PCA. These kinds of systems can track pupils very robustly eliminating the need of manual initialization or any kind of pre-processing.

The tracking of facial features in detail requires recovering the parameters that drive the shape of the feature. A lot of research has been directed towards recovering shape using image matching techniques. Jones and Poggio [22] used a stochastic

gradient-descent based technique to recover the shape and the texture parameters. Cootes et al [8] use active appearance models, which are statistical appearance models generated by combining a model of shape variation with a model of texture variation, for the purpose of image matching. Covell et al [10, 11] exploit the coupling between the shape and the texture parameters using example images labeled with control points. The shape is recovered from the image appearance in an non-iterative way using eigen analysis. Given an image of the face, this approach first needs to estimate the location of features of interest, which is difficult to do robustly in real time.

Most of the approaches mentioned are not highly robust as often these problems need to solve a search problem in this very high dimensional space. Although there are techniques to reduce the dimensionality [32], it is mostly impossible to span the space. Lately there has been a lot of interest in solving vision problems using belief propagation [19, 33]. Rather than analyzing the whole image, small patches (hence low dimensional) are analyzed and belief about each patch is propagated in the whole image. Coughlan et al [9] have used belief propagation to fit deformable shape models. Freeman et al [19] have provided a framework for solving low level vision problems using belief propagation and showed some promising results.

Chapter 3

System Overview

To recognize facial actions, relevant features from the observations need to be extracted. The observation here can be raw video or any other sensory information obtained from the face. Once the features are extracted, the second task is to classify them to recognize different facial actions. The performance of the recognition task depends not only on how well the features represent the facial actions, but also on how well these features can be extracted.

Most researchers have focused on video or images of the frontal face [1, 13, 18, 31, 3]. The features that are usually extracted for the purpose of face analysis range from optical flow fields [4, 18, 3] to gabor coefficients [37]. A lot of different approaches [4, 13, 18, 31, 3, 37] have been used to classify the features to recognize facial actions.

This work is divided into three parts. Figure 3-1 gives you an overview of the system. The first two parts are concerned with robust extraction of the features that are highly correlated with the facial actions. I use more than just computer vision to extract these features. The red-eye effect [21] is a a physiological property of the eye and the first part concerns using it to robustly track the pupils. Once the pupil positions are known, those are then used to get the shape information of the eyes and the eyebrows using templates. Finally, the upper facial action units and head gestures are recognized using machine learning techniques that classify the extracted features. This chapter explains the overall architecture of the system in detail with emphasis on the first part.

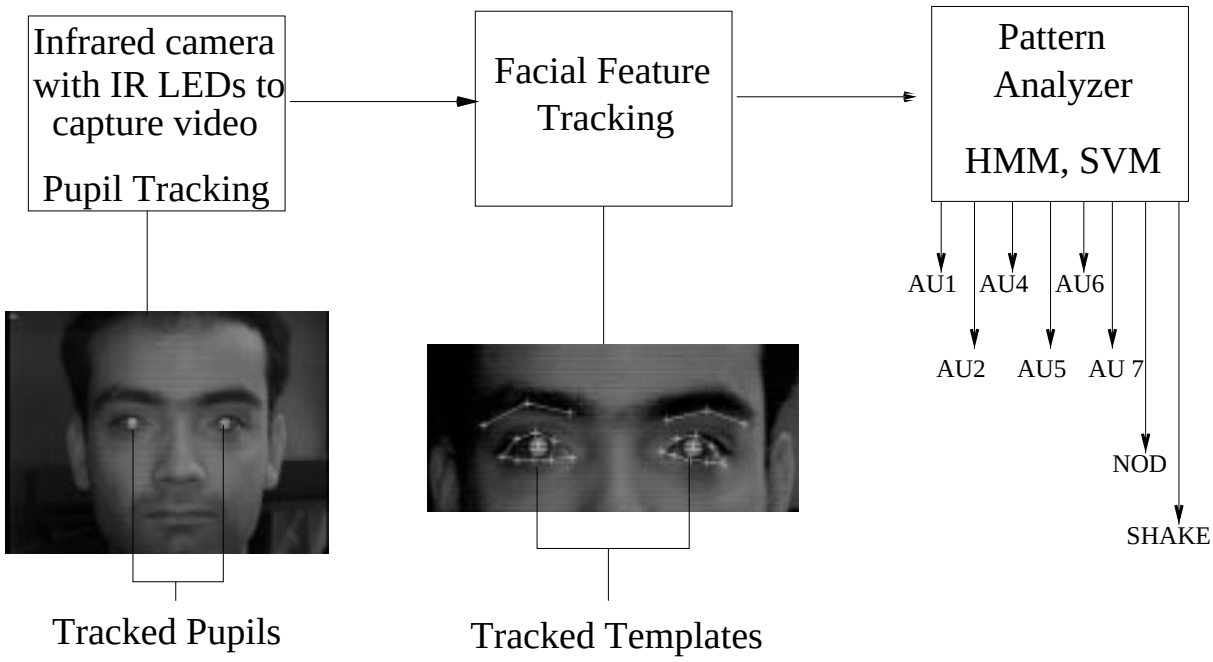

Figure 3-1: The overall system

3.1 Part 1: Finding and Tracking Pupils

In the foundation of this work, lies a system that can detect pupils using the red-eye effect. The system's robustness to occlusions and head motions makes it ideal to be used for facial feature extraction. As the pupil positions can be recovered very efficiently and robustly, it eliminates the need of manual labeling or pre-processing of the images, a required step that plagues a number of pure vision based approaches.

Although the red-eye effect has been known for quite sometime, it is in recent years that it has grabbed a lot of attention for vision applications. Morimoto et al [25] have described a system to detect and track pupils using the red-eye effect. Haro et al [21] have extended this system to detect and track the pupils using a Kalman filter and probabilistic PCA. I use an infrared camera equipped with infrared LEDs, which is used to highlight and track pupils and is an in-house built version of the IBM Blue Eyes camera (http://www.almaden.ibm.com/cs/blueeyes).

The pupil tracking system is shown in Figure 3-2. The whole unit is placed under the monitor pointing towards the users face. The system has an infrared sensitive

Figure 3-2: Camera to track pupils, placed under the monitor

camera coupled with two concentric rings of infrared LEDs. One set of LEDs is on the optical axis and produces the red-eye effect. The other set of LEDs, which are off axis, keeps the scene at about the same illumination. The two sets of LEDs are synchronized with the camera and are switched on and off to generate two interlaced images for a single frame. The image where the on-axis LEDs are on has white pupils whereas the image where the off-axis LEDs are on has black pupils. These two images are subtracted to get a difference image, which is used to track the pupils. Figure 3-3 shows a sample image, the de-interlaced images and the difference image obtained using the system.

The pupils are detected and tracked using the difference image, which is noisy due to the interlacing and motion artifacts. Also, objects like glasses and earrings can show up as bright spots in the difference image due to their specularity. To remove this noise we first threshold the difference image using an adaptive thresholding algorithm [21]. First, the algorithm computes the histogram and then thresholds the image keeping only 0.1 % of the brightest pixels. All the non-zero pixels in the resulting image are set to 255 (maxval). The thresholded image is used to detect and to track the pupils. The pupil tracker is either in a detection mode or a tracking mode. Whenever there is information about the pupils in the previous frame the tracker is in tracking mode and whenever the previous frame has no information about the pupils the tracker switches to the detection mode. The pupil tracking algorithm is shown in Figure 3-4.

During the tracking mode the tracker maintains a state vector, comprised of the

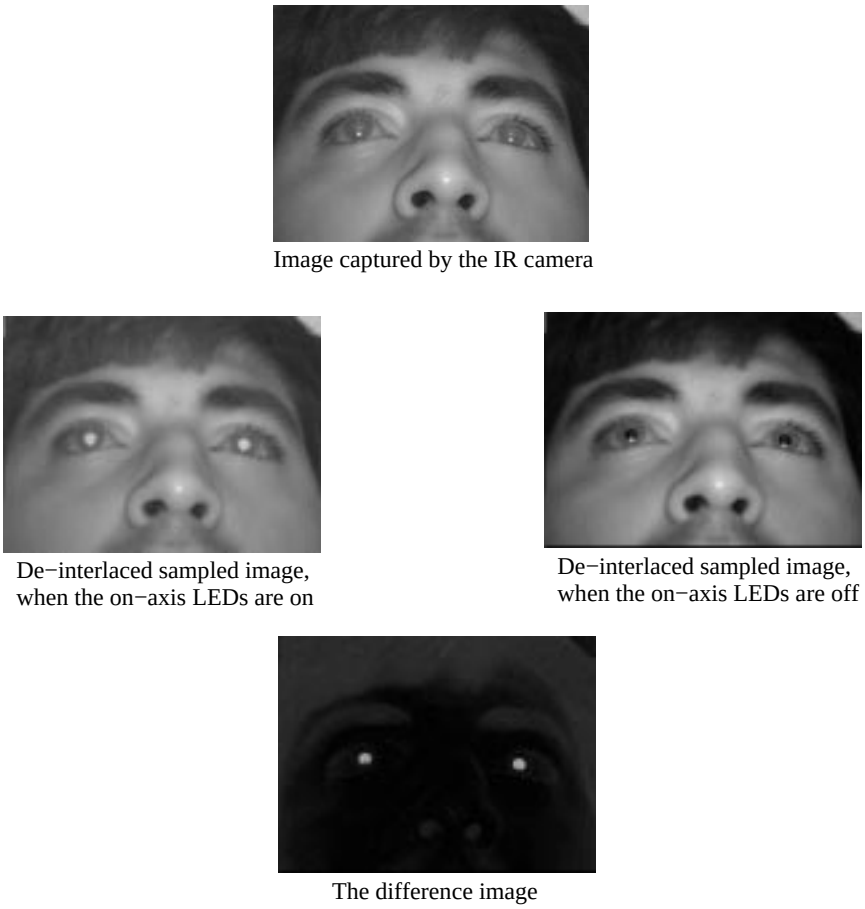

Figure 3-3: Pupil tracking using the infrared camera

spatial information about the pupils. Specifically, the average distance between the pupils during the current tracking phase and their x, y coordinates in the previous frames is maintained. To obtain the new positions of pupils a search for the largest connected component is limited to a bounding box centered on previous pupils. The new connected components are accepted as valid pupils when they satisfy a number of spatial constraints. If the area is greater and the displacement of their centers from previous pupil position lies below a certain threshold, the connected components are considered valid. Also if a connected component is found for both the eyes then the distance between these pupils is also compared with the average distance maintained in the state space to rule out false detections. Once the connected components are identified as valid pupil regions, the state vector is updated.

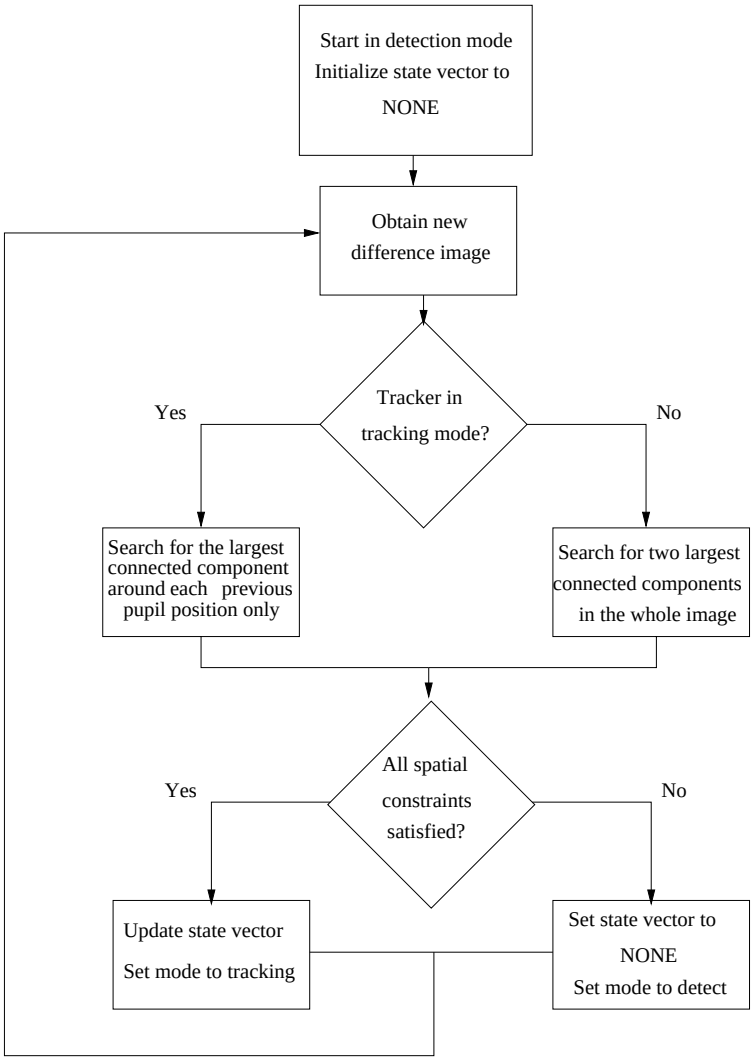

Figure 3-4: The Pupil tracking Algorithm

The tracker switches to the detection mode whenever there is no information about the pupils. In this mode the tracker simply selects the two largest connected components that have an area greater then a certain threshold. Again, to validate the regions, we apply some spatial constraints. Head movements during head nods and head shakes do produce motion artifacts but due to the nature of our algorithm to spatially constrain the search space, it tracks the pupils well. In extreme cases when head movements are too fast, the pupils are lost as motion artifacts overpower the red-eye effect and the pupils are absent from the difference image altogether.

3.2 Part 2: Finding and Tracking Facial Features

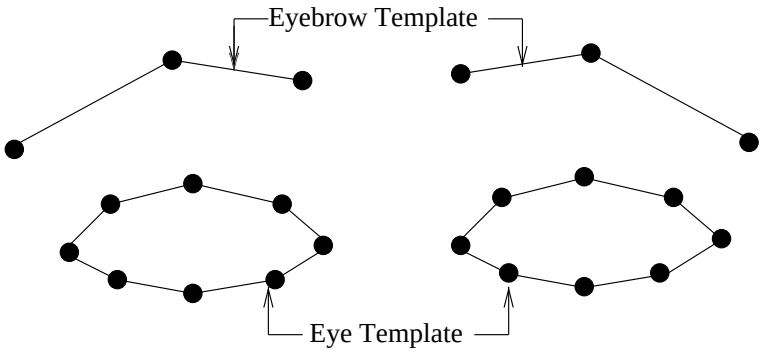

Figure 3-5: Eye and Eyebrow Templates

Templates are used to represent the detailed shape information of the facial features. Eye and eyebrow templates, used in the system, are shown in Figure 3-5. A set of 8 points placed on the eye contour describes the shape of an eye. Two of these points correspond to the eye corners and the rest are equidistant on the contour. Similarly 3 points are used to describe the shape of an eyebrow. A total of 22 points (8 for each eye and 3 for each eyebrow) are used to describe the positions and shapes of upper facial features. Tian et al [29, 30] use a template that consists of the iris location and two parameterized parabolas, which would fit the lower and the upper eyelid. They use the corners of eyes and center points on the eyelids to extract the template parameters. As our system tracks more points than eye-corners and center points on the eyelid, it can represent more detailed and accurate information about the shape.

The pupil positions obtained in part 1 are used to crop out *regions of interest*: two 140 x 80 pixel images of the eyes and two 170 x 80 pixel images of eyebrows. The template parameters for the features are recovered by analyzing these extracted images using example based learning. Chapter 4 describes how to recover these shape parameters in detail.

3.3 Part 3: Classifying Facial Actions

Once the parameters describing the facial features are recovered, the facial actions and head gestures are recognized using machine learning techniques. A Hidden Markov Model (HMM) [26] based classifier is used to detect head nods and head shakes. A separate Support Vector Machine is trained for each facial action unit. The parameters that describe the shape of facial feature in a frame are first normalized to account for different head orientations. These normalized parameters are used as input features to the support vector machines to detect occurrence of facial actions. Since we use a separate classifier for each action unit, they can detect action unit combinations. Chapter 5 describes the classification of facial action and head gestures in greater detail.

Chapter 4

To Recover the Shape

For the purpose of facial action analysis, we need to track the facial features robustly and efficiently. Also, rather then just tracking the positions of facial features, we need to recover the parameters that drive the shape of the feature. The variability in appearance of facial features changes due to pose, lighting, facial expressions etc making the task difficult and complex. Even harder is the task of tracking the facial features robustly in real time, without any manual alignment or calibration. Many previous approaches have focused just on tracking the location of the facial features or require some manual initialization/intervention. In this chapter, I describe how we can robustly recover the shape of facial features in detail using templates in real time without requiring any manual intervention.

Our system exploits the fact that it can estimate the location of pupils very robustly in the image. Once the pupils are located, *regions of interest* corresponding to eyes and eyebrows are cropped out and analyzed to recover the shape description. For the purpose of the facial action analysis, the fiducial points of the templates describing eyes and eyebrows(Figure 3-5) are considered as shape parameters. Our goal is then to recover these fiducial points in a new image.

Cootes et al [8] distinguishes two kinds of parameters that characterize an image. Those, that correspond to the texture of an image and, those that drive the shape of the object of interest. Jones et al [22] similarly represent an image in terms of a shape vector and a texture vector. Although the texture parameters can be recovered fairly

easily, it turns out that the shape parameters introduce a non-linearity in the search for parameters. Techniques ranging from optical flow [2] to gradient descent [8, 22] have been suggested to recover the shape. The methods described in this chapter use learning by example. Specifically, given some example images as training data, hand annotated for shape parameters, the techniques tries to estimate the relationship between the shape parameters and the example images. The shape in a new image then can be estimated using this learnt relationship.

The next section describes a very simple approach that recovers shape parameters assuming a linear relationship between shape parameters and image observations. This method is most similar to the eigenpoints approach by Covell et al [10, 11]. Following that, the same problem is discussed in a more general Bayesian estimation framework. After that, I describe how belief propagation can be used to further refine the shape recovery and address some issues that arise in the Bayesian estimation framework. Finally, I conclude the chapter with some experimental results. For all the methods described in this chapter it is assumed that we have a training set of image vectors $\{\mathbf{i}_1, \mathbf{i}_2, .., \mathbf{i}_n\}$, where each image vector $\mathbf{i}_k$ is pre-annotated with a corresponding vector of shape parameters $\mathbf{s}_k$. For the purpose of facial action analysis, the images $\mathbf{i}$ are cropped images of eyes and eyebrows and the vector of shape parameters $\mathbf{s}$ is a stack of x,y coordinates of fiducial points.

4.1 Recovering Shape using Principal Component Analysis

To recover the shape parameters in a test image, say $\mathbf{i}_{test}$, a very naive approach will be to find an image, $\mathbf{i}_{match}$, from the training set of pre-annotated images that resembles most to $\mathbf{i}_{test}$. The shape parameters of $\mathbf{i}_{test}$ then can be approximated by the shape parameters $\mathbf{s}_{match}$, which corresponds to $\mathbf{i}_{match}$. This approach cannot generalize well, as there can be only a finite number of example images in the training database. A more general approach will be to represent the test image as a linear

combination of example images. The same linear combination can be applied to the corresponding shape parameters of the example images to recover the shape in the new image. Principal component analysis (PCA) can be used to figure out the representation of the test image in terms of the linear combination of example images. Given n example images $\mathbf{i}_k$, let $\mathbf{s}_k$ $(k = 1..n)$ be vectors corresponding to the marked control points on each image. If $\bar{\mathbf{i}}$ is the mean image, then the covariance matrix of the training images can be expressed as:

$$\mathbf{\Lambda} = \mathbf{P}\cdot\mathbf{P}^T \text{ where } \mathbf{P} = [\mathbf{i}_1 - \bar{\mathbf{i}}, \mathbf{i}_2 - \bar{\mathbf{i}}, ..., \mathbf{i}_n - \bar{\mathbf{i}}]$$

The eigenvectors of $\mathbf{\Lambda}$ can be computed by first computing the eigenvectors for $\mathbf{P}^T\cdot\mathbf{P}$. If $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..\mathbf{v}_n]$ where $\mathbf{v}_k$ represents the eigenvectors of $\mathbf{P}^T\cdot\mathbf{P}$, then the eigenvectors $\mathbf{u}_k$ of $\mathbf{\Lambda}$ can be computed as:

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n] = \mathbf{P}\cdot\mathbf{V}$$

As the eigenvectors are expressed as a linear combination of example images, we can express the shape parameters corresponding to the eigen images using the same linear combination. Let $\bar{\mathbf{s}}$ be the mean of the vectors corresponding to the control points in example images and let $\mathbf{Q} = [\mathbf{s}_1 - \bar{\mathbf{s}}, ..., \mathbf{s}_n - \bar{\mathbf{s}}]$, be the matrix of unbiased shape parameters. Then, the shape parameters $\tilde{\mathbf{s}}_k$ $(k = 1..n)$ corresponding to an eigenvector $\mathbf{u}_k$ can be computed as:

$$[\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, ..., \tilde{\mathbf{s}}_n] = \mathbf{Q}\cdot\mathbf{V}$$

To recover the shape parameters in the test image, we first express the new image as a linear combination of the eigenvectors by projecting it onto the top few eigenvectors.

$$\mathbf{i}_{new} = \sum_k a_k\mathbf{u}_k + \bar{\mathbf{i}} \tag{4.1}$$

where $a_k = (\mathbf{i}_{test} - \bar{\mathbf{i}})^T\cdot\mathbf{u}_k$ and $\mathbf{u}_k$ is the k^{th} eigenvector.The same linear combination is applied to the shape parameters of corresponding eigenvectors to recover the new shape.

$$\mathbf{s}_{new} = \sum_k a_k \tilde{\mathbf{s}}_k + \overline{\mathbf{s}} \tag{4.2}$$

This strategy is a simplification of the approach used by Covell et al[10, 11] and performs well for the purpose of the facial feature tracking, particularly on the subjects who had images in the training set. Note that there is no initialization step, which was very critical in many template matching approaches. Further, the non-iterative nature of the approach makes it ideal to be used in a real-time system.

Despite various advantages, this strategy has some shortcomings. It assumes a linear relationship between the image and the shape parameters, which might not be the case always. Also, it uses principal component analysis to recover the shape, hence it inherently assumes that the top eigenvectors capture the shape variations, which is erroneous. There may be variations due to lighting which would contribute highly to the principal components. As described in the next section, this strategy is a special case of a Bayesian estimation framework and we can come up with a method to recover shapes that does not have these problems.

4.2 Recovering Shape in an Estimation Framework

The general scenario of Bayesian estimation is that there is some measurement $\mathbf{y}$ and some unobserved quantity of interest $\mathbf{x}$. Given joint statistics of $\mathbf{x}$ and $\mathbf{y}$, we want to find out the function that best estimates $\mathbf{x}$ based on observing $\mathbf{y}$. In a Bayesian framework this estimator function is chosen to optimize a suitable performance criterion. In particular, given a cost function $C(\mathbf{a}, \hat{\mathbf{a}})$ that specifies the cost of estimating a vector $\mathbf{a}$ as $\hat{\mathbf{a}}$, we choose the estimator that minimizes the average cost, i.e.,

$$\hat{x}(\cdot) = \arg\min_{\mathbf{f}(\cdot)} E[C(\mathbf{x}, \mathbf{f}(\mathbf{y}))] \tag{4.3}$$

Given a suitable cost criterion we can find out an optimal estimator $\hat{x}(\cdot)$ in a Bayesian sense. One of the most commonly used cost criterion is the least square error. If we choose least square error as our error criterion then the estimator is the

mean of the posterior density $p_{\mathbf{x}|\mathbf{y}}(x|y)$ and is called the Bayes least square estimator.

$$\hat{x}_{bls}(y) = E[\mathbf{x}|\mathbf{y} = y] = \int_{\mathbf{x}} x p_{\mathbf{x}|\mathbf{y}}(x|y) d\mathbf{x} \tag{4.4}$$

We can use this Bayes least square estimate to recover the shape parameters given an image. So in terms of the variables used above, the measurement $\mathbf{y}$ is a random vector $\mathbf{I}$ corresponding to the observed image and the quantity of interest $\mathbf{x}$ is a random vector $\mathbf{S}$ corresponding to the shape parameters. If we know the posterior $p_{\mathbf{S}|\mathbf{I}}(\mathbf{s}|\mathbf{i})$, then given an image vector $\mathbf{i}$ we can estimate its shape parameters in a Bayes least square sense according to equation 4.4. Critical to this estimation is the posterior, which can be determined using the Bayes rule as:

$$p_{\mathbf{S}|\mathbf{I}}(\mathbf{s}|\mathbf{i}) = \frac{p_{\mathbf{S},\mathbf{I}}(\mathbf{s},\mathbf{i})}{p_{\mathbf{I}}(\mathbf{i})}$$

Given example images pre-annotated for shapes, the posterior can be estimated using regular probability density modeling techniques. Due to the huge dimensionality of the image space, it is almost impossible to accurately estimate the posterior probability density using just a few hundred example images. The problem becomes simpler if we assume certain properties about the distribution. For example, a lot of people [32, 11] have tried to model image spaces using principal component analysis (PCA), which by virtue of considering only mean and covariance is equivalent to assuming a Gaussian model. These assumptions on the probability densities constrain the relationship between image and the shape. One special case is when the shape parameters $\mathbf{S}$ and the image observations $\mathbf{I}$ are jointly Gaussian. For the jointly Gaussian case the Bayes least square estimate is equal to the linear least square estimate and can be computed as,

$$\hat{\mathbf{s}}_{bls}(\mathbf{i}) = E[\mathbf{S}|\mathbf{I} = \mathbf{i}] = \overline{\mathbf{s}} + \mathbf{\Lambda}_{\mathbf{S},\mathbf{I}} \mathbf{\Lambda}_{\mathbf{I}}^{-1} (\mathbf{i} - \overline{\mathbf{i}}) \tag{4.5}$$

$$\text{Where } \overline{\mathbf{s}} = mean(\mathbf{S}),\ \overline{\mathbf{i}} = mean(\mathbf{I}),$$
$$\Lambda_{\mathbf{S},\mathbf{I}} = Covariance(\mathbf{S},\mathbf{I}) \text{ and } \Lambda_{\mathbf{I}} = Variance(\mathbf{I})$$

Now, consider the combined image and shape parameter matrix $[\mathbf{P}^T\ \mathbf{Q}^T]^T$. $\mathbf{P}$ and $\mathbf{Q}$ are unbiased matrices of image data and corresponding shape parameters respectively. Each column of $\mathbf{P}$ corresponds to an unbiased image represented as a column vector. Similarly, each column of $\mathbf{Q}$ corresponds to an unbiased vector of shape parameters. In the combined matrix each column of $\mathbf{P}$ is aligned with it's corresponding shape vector in $\mathbf{Q}$. If we consider the following eigen decomposition [10, 11],

$$\begin{bmatrix}\mathbf{P}\\ \mathbf{Q}\end{bmatrix}\begin{bmatrix}\mathbf{P}\\ \mathbf{Q}\end{bmatrix}^T = \begin{bmatrix}\mathbf{U_I}\\ \mathbf{U_S}\end{bmatrix}\mathbf{\Sigma}^2\begin{bmatrix}\mathbf{U_I}\\ \mathbf{U_S}\end{bmatrix}^T \tag{4.6}$$

Columns of $\left[\begin{smallmatrix}\mathbf{U_I}\\ \mathbf{U_S}\end{smallmatrix}\right]$ are eigen vectors of the combined image shape parameter subspace. It can be shown that $\mathbf{\Lambda_{S,I}} = \mathbf{U_S}\ (c\mathbf{\Sigma}^2)\ \mathbf{U_I}^T$ and $\mathbf{\Lambda_I} = \mathbf{U_I}\ (c\mathbf{\Sigma}^2)\ \mathbf{U_I}^T$. Where c is a scalor and equal to $\frac{1}{\text{number of examples} - 1}$. Plugging in for $\mathbf{\Lambda_{S,I}}$ and $\mathbf{\Lambda_I}$ in equation 4.5 we get,

$$\hat{\mathbf{s}}_{bls}(\mathbf{i}) = \overline{\mathbf{s}} + \mathbf{U_S}\mathbf{U_I}^{-1}(\mathbf{i} - \overline{\mathbf{i}}) \tag{4.7}$$

Comparing this equation with equation 4.2, we can see that both the methods are very similar. In fact the two methods are equivalent if the columns of $\mathbf{U_I}$ are exactly equal to the eigen vectors of the image space. So, if we assume $\mathbf{S}$ and $\mathbf{I}$ to be jointly Gaussian then this Bayesian approach reduces to the approach described earlier and constrains the relationship between $\mathbf{S}$ and $\mathbf{I}$ to be linear. Also this means that estimation in this framework will suffer from the same drawbacks of the previous approach. So rather than making assumptions about the probability densities, perhaps, we should try to model the posterior as exactly as possible. There are modeling techniques like sampling and representations like mixture of Gaussian, that allow to model any kind of probability density function. Unfortunately, it is very difficult to model this probability density function accurately. The dimensionality of image space is very high and compared to this dimensionality the number of example images is small.

Since working with the whole image is so difficult, it might be better to work with smaller, more manageable image patches. The whole image can be broken down to small, manageable image patches and can be analyzed to recover some local beliefs. Then to recover a global property of the image (for example, shape parameters) the local beliefs can be juxtaposed and analyzed in totality. This is the main idea behind recovering shape parameters using belief propagation, which we describe in the next section.

4.3 Recovering Shape by Propagating Bayesian Beliefs

In recent years, a number of researchers have used belief propagation to solve a number of problems in computer vision [9, 19, 33]. This method is motivated by VISTA, Vision by Image/Scene TrAining, as described by Freeman et al [19]. The relationship between an image and its shape parameters can be modeled by first estimating a relationship between local image patches and shape parameters, and then modeling the relationship between the shape parameters of neighboring image patches. Figure 4-1 shows the graphical way to represent this relationship. The whole image $\mathbf{I}$ is divided into image patches represented by y_i's. The x_i's, underlying scene variables correspond to the shape parameters, are statistically related to an image patch y_i and also to its spatial neighbors. The link between y_i's and x_i's allows an initial scene estimate, whereas the link between an x_i and its spatial neighbors allows the estimate to propagate. Under the Markovian assumptions, the underlying scene variable (x_i), given values of all its neighboring scene variables, should contain all the information required to recover shape parameters from the image patch it connects to.

Consider the case when the shape parameter to be recovered is only one fiducial point on the image. Then, the underlying scene variables can be assigned values according to their spatial distribution with respect to the fiducial point in the image.

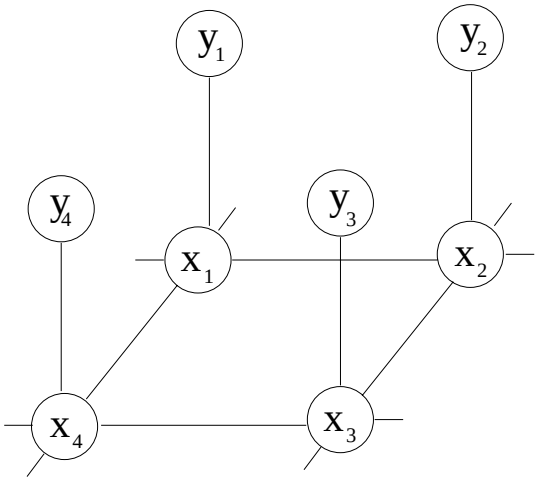

Figure 4-1: Markov network topology to recover the shape

Figure 4-2 shows how the scene variables are assigned based on their spatial positions. The whole image is divided into patches (shown by dotted lines). We use 20 x 20 image patches in our implementation. The shape parameter is a single fiducial point which lies in one of the image patches. The scene variable corresponding to the image patch where the fiducial point lies takes the value $(same, same)$. Similarly, the scene variable corresponding to other image patches can take one of the nine possible states, depending upon the spatial position of the image patch they are connected to. This method works as if it was solving a jigsaw puzzle. The patches are analyzed individually to get the local beliefs. The aim here is to find a configuration that will both support the local evidence and satisfy the spatial constraints as well. Once the beliefs are propagated, the image patch most likely to be in state $(same, same)$ is chosen and the fiducial point can be recovered using the techniques similar to the ones described earlier in the chapter. The difference is that rather than analyzing the whole image, we only need to analyze the patches most likely to contain the fiducial points. Further, we can more accurately model the probability distribution between image patches and the shape parameters.

The approach described here is mostly limited to the case where the shape parameter is just one fiducial point. For scenarios where the shape description has more

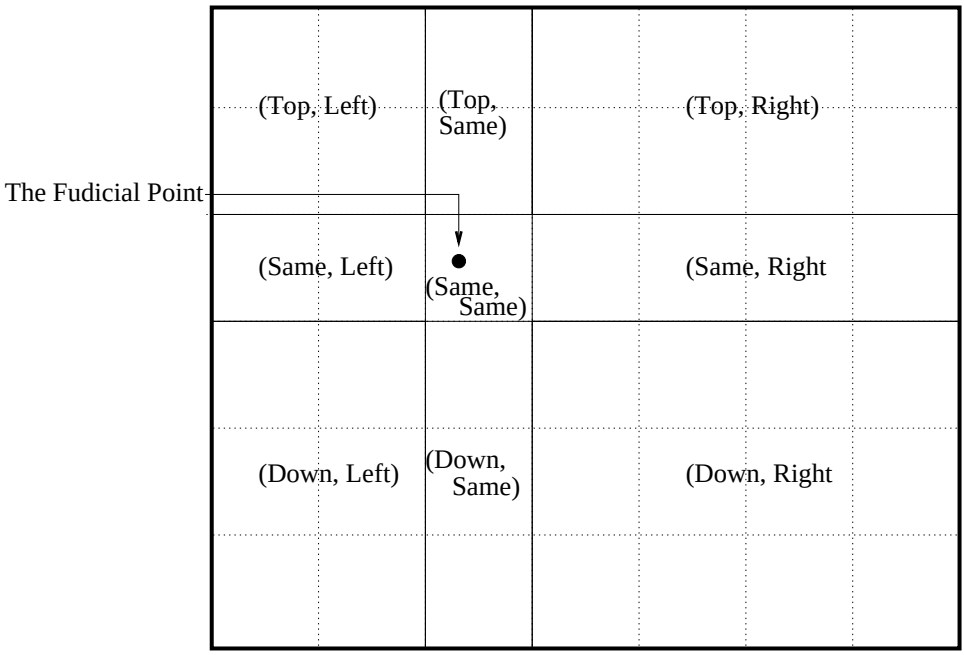

Figure 4-2: Spatial distribution of scene variables

than one point, this approach can be applied to each point separately. The choice of representation for the underlying scene variable is very critical to the performance of this approach and perhaps this representation might not be the best.

The statistical relationships between image patches and scene variables and between neighboring scene variables can be learnt using examples in the training set. To recover the shape, the patches of the new image can be analyzed and beliefs about the state of the underlying scene variables can be propagated using inference algorithms. One such inference algorithm is belief propagation. If we denote the relationship between y_i's and x_i's as $\phi(y_i, x_i)$, and the relationship between neighboring nodes (i, j) as $\psi(x_i, x_j)$ then the joint probability between the scene variables and the image can be given by [19],

$$P(x_1, x_2, ..., y_1, y_2, ..) = \prod_{(i,j)} \psi(x_i, x_j) \prod_k \phi(y_k, x_k)$$

In our implementation, the compatibility functions ψ and ϕ are conditional probabilities: $\psi(x_i, x_j) = P(x_i|x_j)$ and $\phi(y_j, x_j) = P(x_j|y_j)$. These can be computed from the training data. The Bayes least square estimate for $\hat{x}_i$ can be computed by

marginalizing over the other variables and for discrete variables can be computed as:

$$\hat{x}_{j_{bls}} = \sum_{x_j} x_j \sum_{x_i, i \neq j} P(x_1, x_2, ..., y_1, y_2, ..)$$

The Bayes least square estimate can be computed for networks without loops using simple message passing rules as following [19]:

$$\hat{x}_{j_{bls}} = \sum_{x_j} x_j \phi(y_j, x_j) \prod_k M_j^k \tag{4.8}$$

Here k runs over all the neighbors of j. if $\tilde{M}_k^l$ is the message from node l to k in the previous iteration then M_j^k, the message from node k to node j, can be computed as:

$$M_j^k = \sum_{x_k} \psi(x_j, x_k) \phi(y_k, x_k) \prod_{l \neq j} \tilde{M}_k^l \tag{4.9}$$

The Markov network suggested for shape recovery contains loops and unfortunately the message passing rules mentioned above are invalid for networks with loops. But there are strong experimental and theoretical results [19, 35] that motivate application of belief propagation rules.

4.4 Implementation and Results

For all the tests described here, our training set consisted of 150 images of eyes and eyebrows from ten different individuals with different facial expressions and different lighting conditions. Each eye region of size 140 x 80 is separately analyzed. Similarly each eyebrow image of 170 x 80 pixel resolution is used to recover the eyebrow template. These images were hand marked to fit the facial feature templates.

The approach that uses PCA to recover shape was implemented and worked in real time at 30 fps on a Pentium-III 933 MHz Linux machine. The training set was first processed offline to compute the required eigenvectors. During the real-time tracking the cropped images of the eyes and eyebrows are projected on the corresponding top

Table 4.1: Mean RMS error per control point location

	PCA	**Belief Propagation**
Subject in Training Set	0.65	0.80
Subject not in Training Set	0.78	0.83

few eigenvectors. Experiments showed that first 40 eigenvectors were good enough for the task and in our implementation we use those. These projections are used to recover the control points using the approach explained before. This approach worked particularly well on the subjects who had their images in the training database.

The approach that uses belief propagation to recover shape was also implemented. The system was not real-time. The training data was used to learn the compatibility function ψ and ϕ. For simplicity, PCA based shape recovery was used to recover the fiducial point from the image patch most likely to contain the point.

Both the approaches were compared on two image sequences. The first sequence consisted of 93 frames and the subject in that sequence was in the training database. The second sequence was 100 frames long and the subject in this sequence did not appear in the database. The eye and eyebrow corners were handmarked in both the sequences and these points were compared with the points tracked automatically by both the approaches. Table 4.1 shows the mean RMS difference per control point between the points manually marked and points tracked by the two approaches. As shown by results, the belief propagation as implemented here does not provide significantly better results as compared to the PCA based approach. Our real-time system tracks the facial features using the PCA based approach, but the belief propagation seems promising and we are further exploring better techniques that will enable us to track the facial features more accurately.

Figures 4-3 and 4-4 show tracking results of some sequences. Both the subjects appearing in Figure 4-3 were in the training database. The system is able to track the features very well. Note that in the first sequence of Figure 4-3 the left eyebrow is not tracked in frames 67, 70 and 75 as it is not present in the image. Similarly all the templates are lost in the frame 29 in the second sequence of Figure 4-3 when the

pupils are absent, as the subject blinks. The templates are recovered as soon as the features reappear. Figure 4-4 shows the tracking results for the subjects not in the training set. Again, note that the second frame in the first sequence does not show any eyes or eyebrows, due to the fact that the subject blinked and hence no pupils were detected. The tracking is recovered in the very next frame when the pupils are visible again.

The results show that the system is very efficient, runs in real time at 30 fps and is able to track upper facial features robustly in presence of large head motions and occlusions. One limitation of our implementation is that it is not invariant to large zooming in or out as fixed size images of the facial features are cropped. Further, our training set did not have samples with scale changes. Also in a few cases with some new subjects, the system did not work well, as the training images were not able to span the whole range of variations in appearance of the individuals. A training set which captures the variations in appearance should be able to overcome these problems.

Frame 57 Frame 67 Frame 70 Frame 75 Frame 88

Frame 27 Frame 28 Frame 29 Frame 30 Frame 31

Figure 4-3: Tracking results for subjects in training set

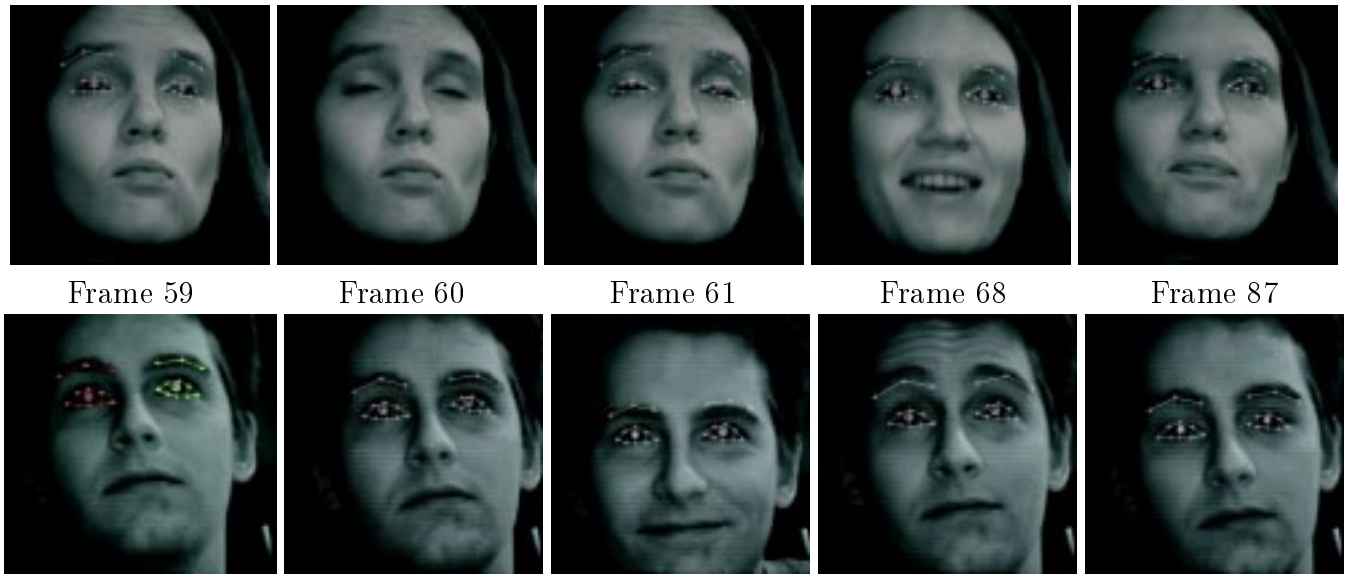

Figure 4-4: Tracking results for subjects not in training set.

Chapter 5

Recognizing Facial Actions

Once the shape parameters describing the facial features are extracted the next step is to identify the facial actions they correspond to. The head gestures can also be recognized by observing these facial parameters over time. In this chapter, I discuss the challenges involved in building a real-time system that recognizes head gestures and upper facial action units (AUs). The first section concerns the facial action recognition from a video frame. Following that section, I describe how temporal information can be used to recognize head gestures in real time.

5.1 Facial Action Recognition

There are over 40 different facial action units enumerated in FACS [17] and more than 7,000 different AU combinations have been observed. A system that aims to analyze faces should not only recognize a single AU but the combinations of AUs as well. The AU combinations can be additive or non-additive. The appearance of AUs does not change when additive combination of AUs occur, whereas in non-additive combinations, the appearance of individual AUs does change.

Researchers in the past have used a number of classification techniques to recognize action units and their combinations. Donato et al [13] have shown classification results based on a number of feature extraction techniques. They have used a nearest neighbor classifier and template matching for the purpose of recognition. Each facial

Table 5.1: Shape parameters used for recognizing AUs

Action unit	Facial action	Shape Parameters Used
1	Inner brow raiser	Fiducial points on eyebrows
2	Outer brow raiser	Fiducial points on eyebrows
4	Brow lowerer	Fiducial points on eyebrows
5	Upper eye lid raiser	Fiducial points on eyes
6	Cheek raiser	All fiducial points
7	Lid tightener	Fiducial points on eyes
	Neutral	All fiducial points

action combination that they try to recognize is treated as a separate AU. As there are a large number of AU combinations, modeling each AU combination separately is not appropriate. Tian et al [31] have used neural networks to recognize facial actions and their combinations. They use a separate neural network for the upper face and for the lower face, and given all the upper or lower facial feature parameters as input, multiple nodes corresponding to each occurred AU are excited.

There are lots of classifiers that could be used for the purpose of AU recognition. Support Vector Machines (SVM) have been shown to perform well on a number of classification tasks. SVM is an optimal discriminant method based on Bayesian learning theory and generalizes well. I use Cawley's SVM toolbox [5] to train the SVMs to classify facial feature parameters that correspond to an occurrence of a particular AU from the ones that don't. A separate SVM for each AU is trained using examples. During the recognition phase, the extracted facial feature parameters in each frame are classified by all the SVMs to figure out which AUs were present. Also rather than using all the shape parameters, we use only those that are most indicative of the action unit that we are trying to recognize. Table 5.1 shows the parameters used to recognize each action unit.

5.1.1 Classification using Support Vector Machine

Classifiers based on support vector machine (SVM) perform binary classification by first projecting the data points onto a linearly separable feature space and then, using

a hyperplane that is maximally separated from the nearest positive and negative data points. Mathematically, given a set of N training data (x_i, y_i), where $x_i \in \mathbb{R}^d$ with corresponding label $y_i \in \{1, -1\}$, the support vector machine classifies a data point x using,

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i k(x, x_i) - b$$

Here $k(x, x_i)$ is a positive definite kernel function and specifies an inner product between x and x_i in the linearly separable feature space. The x_i's corresponding to non-zero α_i's are support vectors. The α_i's and the bias b can be obtained by solving an optimization problem. For the purpose of classifying facial actions, x is the vector of relevant shape parameters and the sign of $f(x)$ determines whether an AU has been recognized or not.

Before using the facial feature parameters for classification, we need to do some pre-processing. There is a lot of variability in raw facial features due to changes in pose, zoom, personal variations etc. The parameters need to be normalized to account for these variations. Also, rather than using just the facial feature parameter, we should use the relative difference of the facial feature parameters from the parameters that correspond to a neutral face. In the system, we consider the inner eye corners as origins of the coordinate system. The position relative to the eye corner is normalized using the interpupillary distance. These normalized parameters are then subtracted from the normalized parameters corresponding to a neutral frame. These parameters are used as input features to the SVMs. The next section describes evaluation of the system and the results.

5.2 Evaluation and Results

The facial action database has 8 kids in a real learning situation. These kids were asked to play a game called the *fripple place* [14]. The game has a number of puzzles that requires mathematical reasoning. Each kid worked on these puzzles for about 20 minutes. Videos of their faces were recorded by two cameras. A vision camera

Table 5.2: Details of AUs and their combinations in the dataset

AU Combination	Number of Samples
1+2	12
1+2+5	19
1+2+6+7	2
1+4	2
4	10
5	5
7	6
4+7	4
6+7	1
Neutral	19
Total	**80**

Table 5.3: Details of instances of AUs in the dataset

Action Unit	Number of Instances in Database
1	35
2	33
4	16
5	24
6	3
7	13
Neutral	19
Total	**143**

was placed on top of the monitor and an IBM Blue Eyes camera was placed under the monitor. A FACS trained expert coded the videos of the face for various action units and 80 frames were selected from these FACS coded videos of the kids. These frames were selected manually to ensure that there were equal number of samples of the different facial action units from all the kids. Table 5.3 shows the details of the dataset, which contains kids making real facial action units and various combinations.

The facial feature parameters were recovered from all these frames as described in Chapter 4. The system is evaluated for recognition accuracy using leave-one-out cross validation. The classifiers were trained using the data from all but one subject and reserving the one subject for testing. This was repeated for all 8 subjects in the

Table 5.4: Leave-one-out recognition results for the facial actions

Action Unit	Number of Samples	Correct Recognition	Misses	% Correct Recognition
AU 1	35	26	9	74.3%
AU 2	33	26	7	78.8%
AU 4	16	9	7	56.2%
AU 5	24	16	8	66.7%
AU 6	3	0	3	0%
AU 7	13	6	7	46.1%
Neutral	19	14	5	73.3%
Total	143	**97**	**46**	**67.83%**

database. The system could recognize each individual AU with an accuracy of 67.83%, whereas an accuracy of 61.25% was obtained for all AU combinations. Table 5.4 shows how well each individual AU was recognized and Table 5.5 shows how well each AU combination was recognized. Although the results might not sound exceedingly good, we need to keep in mind that these results are reported on a natural dataset; this set is very different from the datasets used to evaluate earlier systems. The videos have a lot of occlusion and head movements, which makes the problem much harder than on datasets where the images are pre-processed and manually normalized.

Table 5.5: Leave-one-out recognition results for action unit combinations

Actual AUs	# of Samples	Fully Recognized	Partially Recognized	Misses	% Full Correct
1+2	12	9	1	2	75%
1+2+5	19	11	3	5	57.9%
1+2+6+7	2	0	2	0	0%
1+4	2	0	2	0	0%
4	10	5	0	5	50%
5	5	5	0	0	100%
7	6	3	0	3	50%
4+7	4	2	1	1	50%
6+7	1	0	0	1	0%
Neutral	19	14	0	5	73.7%
Total	**80**	**49**	**9**	**22**	**61.25%**

5.3 Detecting Head Nods and Head Shakes

Head nods and head shakes are nonverbal gestures and are used to fulfill semantic functions (e.g., nod head instead of saying yes), communicate emotions (e.g., nodding enthusiastically with approval) and as conversational feedback (e.g, to keep the conversation moving). A system that could detect head nods and head shakes would be an important component in an interface that is expected to interact naturally with people. This section describes our approach, which uses Hidden Markov Models (HMMs) [26], to detect head nods and head shakes in real time.

Real-time detection of head nods and shakes is difficult, as the head movements during a nod or shake are small, fast and jerky, causing many video-based face trackers to fail. Our system uses the fact that it can robustly track the pupils. Once the pupil positions are found, they are used to generate observation symbols based on the direction in which the head moved. There are five observation symbols, which correspond to the head moving up, down, left, right, or none. Current pupil positions are compared with pupil positions in the previous frame. If the movement in the x direction is greater than the movement in the y direction then the observation symbol is labeled as left or right head movement depending upon which direction the head moved. Similarly if the movement in the y direction is greater then the movement in the x direction then the label is either up or down, depending upon the direction of the head movement. When the movements in both the x and y directions are below a certain threshold, then the symbol corresponding to none is generated.

The directions of the head movements in consecutive frames are used as a sequence of observations to detect head gestures. Figure 5-1 shows typical patterns associated with the head movements in a nod and a shake. We use a discrete HMM [26] to detect when a head nod or a head shake occurs. Our pattern analyzer consists of two HMMs, one corresponding to head nods and one corresponding to head shakes. Both HMMs have three states and the observation set has five symbols corresponding to the head moving up, down, left, right and none. The HMMs were trained using the Baum Welch algorithm [26] . In the detection phase, the forward-backward procedure

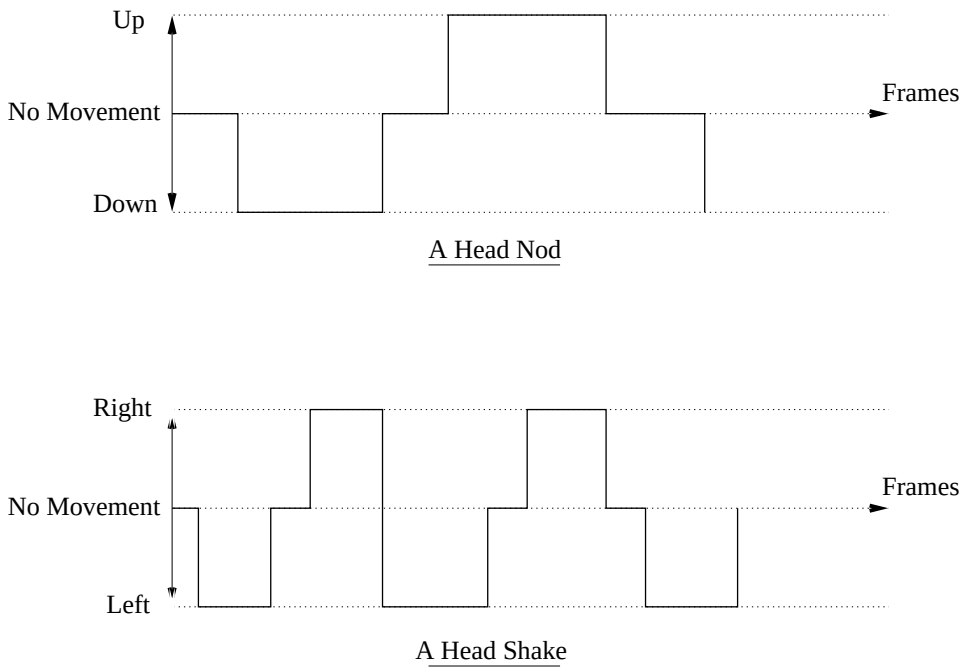

Figure 5-1: Typical sequences of head movements in a head nod and a head shake

[26] is used to compute the log likelihood for a sequence of N consecutive observations based on the two HMMs. We compare and threshold the log likelihood to label the sequence as a head nod or a head shake.

The performance of the system depends upon N, which is the number of observations that constitute a sequence to be tested. If N is small, then slow head nods and shakes might not be detected. When N is large, then the detected head nods and head shakes might linger for some time after they end. Our system uses N=10, which we found sufficient to detect slow as well as subtle head nods/shakes.

5.3.1 Evaluation and Results

Since the facial action database collected for the purpose of facial action recognition had very few nods and shakes we had to build a different database. To collect a natural database for head nods and head shakes a Microsoft agent was programmed to ask a number of factual questions (see Table 5.6), to which the subjects were asked to answer with a head nod or a head shake. We used this strategy to avoid collecting

Table 5.6: Ten questions asked by the agent

1.	Are the instructions clear?
2.	Are you male?
3.	Are you female?
4.	Are you a student at Media Lab?
5.	Are you a student at Boston University?
6.	Were you born in Boston?
7.	Do you like Boston?
8.	Do you like weather here in Boston?
9.	A terrible thing just happened in Nepal recently. Did you hear about it?
10.	[Agent explains the event] Pretty bad isn't it?

data with exaggerated head nods and head shakes, which people often made when asked to just nod/shake their head in front of a camera. Ten subjects, among whom five were male, five female and two of them wore glasses, were recorded using the infrared camera while they interacted with the agent. We expected to have a total of 100 nods and shakes, but there were instances where the subjects responded to a question with nodding/shaking their head twice. Also, some subjects used head nods as conversational feedback to the agent. A total of 110 samples were collected with 62 head nods and 48 head shakes.

Lighting conditions varied due to changes in sunlight coming through a window at different times of day and due to the collection of data from subjects in two different rooms. To further complicate the data, a number of different facial expressions and movements like smiles, and frowns were made by the subjects in addition to the nods and shakes. (Sometimes the agent elicited humor or other responses.) A random 40% of the head nods and 40% of the head shakes were selected for training (see Table 5.7).

The recognition results are shown in Table 5.8 and 5.9. The system was implemented on a Pentium-III 933 MHz Linux machine and a real-time recognition rate of 78.46% was achieved at 30 fps for head nods and head shakes in the test dataset. There were no confusions among head nods and head shakes, as the head movements

in a head nod are very different from those in a head shake. Most of the head nods and head shakes that went undetected were the samples taken from the subjects that wore glasses. The specular nature of the glasses made it difficult for the pupil tracker to work well. Interestingly on one of the subjects with glasses, the pupil tracker tracked a bright specular point on the glass frame and hence was able to detect most of the head nods and head shakes. One of the head shakes that went undetected was because the subject closed his eyes while making the gesture. There were some false positives too. Some head nods were detected when the subject started laughing with the head going up and down rhythmically. Sample demonstration movies can be viewed at http://www.media.mit.edu/~ash/PUI01.

Table 5.7: Number of sequences in training and testing datasets

	Train	Test
Head nods	25	37
Head shakes	20	28

Table 5.8: Recognition results for the training set

	Recognized head nods	Recognized head shakes	Misses
Head nods	23	0	2
Head shakes	0	19	1

Recognition rate for head nods : 92.0%
Recognition rate for head shakes : 95.0%
Combined recognition rate : 93.34%

Table 5.9: Recognition results for the testing set

	Recognized head nods	Recognized head shakes	Misses
Head nods	30	0	7
Head shakes	0	21	7

Recognition rate for head nods : 81.08%
Recognition rate for head shakes : 75.0%
Combined recognition rate : 78.46%

Chapter 6

Conclusion and Future Work

6.1 Summary

This thesis demonstrates a fully automatic, real-time framework that can recognize facial activity and head gestures. This framework can be used in scenarios where the machine needs a perceptual ability to recognize, model and analyze the facial activity in real time without any manual intervention. Rather than trying to recognize specific prototypical emotional expressions like joy, anger, surprise and fear, this system recognizes the head gestures and the upper facial action units enumerated in Paul Ekman's Facial Action Coding System (FACS) [17]. This thesis describes some methods that use statistical learning to first automatically recover parameters describing the facial features, and then use these parameters to recognize facial activity and head gestures. The datasets used for evaluations are completely natural and the thesis demonstrates how computer vision and machine learning can be integrated to build real-world applications.

The system first tracks the pupil positions robustly using the red-eye effect; these positions are then used to localize eyes and eyebrows. The shape parameters corresponding to these facial features are recovered using statistical learning techniques. A real-time system, which uses Principal Component Analysis (PCA) to track upper facial features, is implemented and is shown to work well. Once the parameters describing the facial features are recovered, they are used to recognize the facial actions

and head gestures. Support vector machines (SVMs) are used to recognize facial actions and a recognition accuracy of 67.83% for each individual AU is reported. The system can correctly identify all possible AU combinations with an accuracy of 61.25% in a real and fully natural dataset. The head gestures are recognized using hidden markov models (HMMs) and a recognition accuracy of 78.46% is reported, again on a dataset which was completely natural. A lot of earlier work in face analysis reported very high recognition results and at first glance the results reported here might seem insignificant. But, we have to keep in mind that most of the earlier work has focused on frontal video of the face shot in ideal conditions. The systems were trained and tested at the apex of emotional expression and required human intervention. Considering that an accuracy of 75% among the human FACS coders is considered good, our system performance is comparable to that of humans. In real-world applications, like the learning companion, the face analysis system should be fully automatic and should not require any human intervention, which is challenging due to the presence of head movements, pose variations and occlusions in a natural scenario. This system is evaluated in these challenging conditions and hence, the results reported are the state of the art for natural human-computer interaction.

6.2 Application Scenarios

A system that can recognize facial actions and head movements in real time will find applications in many areas. Some of the possible applications are described below:

- **Learning Companion:** A computerized learning companion needs to detect the underlying affective state of the user in an unobtrusive manner. This system is a part of the effort in the Affective Computing group at the MIT Media Lab to make a system that uses many different behavioral signals to recognize affective states like confusion, boredom etc.

- **Man-Machine Interaction:** The proposed framework can be an important component in a system that aims to interact naturally with people. Applications

that need to be adaptive to the users internal state would find this system useful. The Gestures and Narrative Language group at the MIT Media Lab has started using this system to build synthetic characters that are socially intelligent and use head gestures and eye-gaze as social signals.

- **Behavioral Studies:** A lot of research is focused on studies related to the facial behaviors. This system can be used as a tool that helps to annotate the facial behavior automatically.

6.3 Future Work

The framework suggested in this thesis has several limitations. The system depends heavily upon the robust pupil tracking, which currently breaks when the subjects are wearing glasses. The pattern recognition to find pupils can be further refined to track the pupils even when there are subjects with glasses. Since the system uses infrared LEDs, it is unusable in the presence of a bright infrared source (like the sun) and alternate pupil tracking techniques that do not rely on infrared should be explored. It is also possible to refine the shape parameter extraction by taking into account zoom and variations due to pose changes. Techniques like belief propagation should be further explored to model the relationship between shape parameters and image observations more accurately. Further, the temporal tracking can also be incorporated to make the system more robust. The system can also be extended to track lower facial features, like the lips and nose. In this work, the facial action units are classified using a bunch of separate SVMs. Techniques that fuse different classifiers can be explored to improve the classification. The system can be extended to recognize lower facial action units as well. The video data used in this thesis was collected along with a lot of different signals and it is currently being annotated for affective states like interest and boredom. This whole data needs to be analyzed to figure out what behaviors are correlated with which affective states and based upon this correlation we can build a system that can recognize some emotional states relevant to learning. Currently the results cannot be compared to other existing systems as our tests were performed on

a fully natural database of videos with a lot of occlusions and head movements. For a meaningful comparison all systems should be tested on the same database.

Although in this thesis we have demonstrated a system that can analyze the facial activity, it is still very hard to make machines that can recognize the underlying emotional and cognitive state. The relationship between the facial behavior and the internal state is very complicated and is influenced by many other factors. A system that aims to be socially and emotionally intelligent should also look at factors besides facial behavior. Nonetheless, the face is a very important channel that emits signals related to the internal state and a lot of effort is being devoted to unravel this relationship. Besides being used as a man-machine interface, this framework would hopefully be useful to a lot of these research efforts as well .

Bibliography

[1] M. A. Bartlett, J. C. Hager, P. Ekman, and T. Sejnowski. Measuring facial expressions by computer image analysis. *Psychophysiology*, 36(2):253–263, March 1999.

[2] D. Beymer and T. Poggio. Image representations for visual learning. *Science*, June 1996.

[3] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric model of image motion. In *Proceedings of the International Conference on Computer Vision*, pages 374–381, Cambridge, MA, 1995. IEEE Computer Society.

[4] M. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *Int. Journal of Computer Vision*, 25(1):23–48, 1997.

[5] G. C. Cawley. MATLAB support vector machine toolbox (v0.50β) [http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox]. University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000.

[6] T. Choudhury. Facefacts : Study of facial features for understanding expression. Master's thesis, MIT Media Arts and Sciences, 1999.

[7] N. Chovil. Discourse oriented facial displays in conversation. *Research on Language and Social Interaction*, 25:163–194, 1991.

[8] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence*, 23(6), June 2001.

[9] J. M. coughlan and S. J. Ferreira. Finding deformable shapes using loopy belief propagation. In *Proceedings of European Conference on Computer Vision*, 2002.

[10] Michele Covell. Eigen-points. In *Proceedings of International Conference on Image Processing*, September 1996.

[11] Michele Covell. Eigen-points: control-point location using principal component analyses. In *Proceedings of Conference on Automatic Face and Gesture Recognition*, October 1996.

[12] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 18(1):33–80, January 2001.

[13] G. Donato, M. Bartlett, J. Hager, P. Ekman, and T. Sejnowski. Classifying facial actions. *IEEE Pattern Analysis and Machine Intelligence*, 21(10):974–989, October 1999.

[14] Edmark. Fripple place. http://www.riverdeep.net/edconnect/softwareactivities /critical_thinking/fripple_place.jhtml.

[15] P. Ekman. Facial expression and emotion. *American Psychologist*, pages 384–392, April 1993.

[16] P. Ekman and W. V. Friesen. *Pictures of Facial Affect*. Consulting Psychologist, Palo Alto, CA, 1976.

[17] P. Ekman and W. V. Friesen. *The Facial Action Coding System: A Technique for Measurement of Facial Movement*. Consulting Psychologists Press, San Francisco, CA, 1978.

[18] I. Essa and A. Pentland. Coding, analysis, interpretation and recognition of facial expressions. *Pattern Analysis and Machine Intelligence*, 7:757–763, July 1997.

[19] W. T. Freeman, Pasztor E. C., and O. T. Carmichael. Learning low level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.

[20] J. Hager and P Ekman. Essential behavioral science of the face and gesture that computer scientists need to know. In *International Workshop on Automatic Face and Gesture Recognition*, June 1996.

[21] A. Haro, I. Essa, and M. Flickner. Detecting and tracking eyes by using their physiological properties. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, June 2000.

[22] M. J. Jones and Tomaso Poggio. Multidimensional morphable models. In *Proceedings of International Conference on Computer Vision*, 1998.

[23] S. Kawato and J. Ohya. Real-time detection of nodding and head-shaking by directly detecting and tracking the 'between-eyes'. In *Proceedings of Conference on Automatic Face and Gesture Recognition*, March 2000.

[24] J. Lien, T. Kanade, J. Cohn, and C. C. Li. Detection, tracking and classification of action units in facial expression. *Journal of Robotics and Autonomous Systems*, 31:131–146, 2000.

[25] C. Morimoto, D. Koons, A. Amir, and M. Flickner. Pupil detection and tracking using multiple light sources. Technical report, IBM Almaden Research Center, 1998.

[26] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–284, February 1989.

[27] Johnmarshall Reeve. The face of interest. *Motivation and Emotion*, 17(4), 1993.

[28] J. Scheirer, R. Fernandez, and Rosalind W. Picard. Expression glasses: A wearable device for facial expression recognition. In *Proceedings of Conference on Computer Human Interaction*, February 1999.

[29] Y. Tian, T. Kanade, and J. F. Cohn. Dual-state parametric eye tracking. In *Proceedings of Conference on Automatic Face and Gesture Recognition*, 2000.

[30] Y. Tian, T. Kanade, and J. F. Cohn. Recognizing upper face action units for facial expression analysis. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, June 2000.

[31] Y. Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *Pattern Analysis and Machine Intelligence*, 23(2), February 2001.

[32] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991.

[33] Y. Weiss. Interpreting images by propagating Bayesian beliefs. In M. C. Mozer, M. I. Jordan, and T. Pestche, editors, *Advances in Neural Information Processing Systems*. 1997.

[34] Y. Yacoob and L. Davis. Computing spatio-temporal representation of human faces. In *CVPR*, pages 70–75, Seattle, WA, June 1994.

[35] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. Technical Report 2000-26, MERL, Mitsubishi Electric Research Labs, 2000.

[36] A. Yuille, P. Haallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 1992.

[37] Z. Zhang. Feature-based facial expression recognition: Sensitivity analysis and experiments with a multilayer perceptron. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(6):893–911, 1999.