FACIAL FEATURE TRACKING AND EXPRESSION RECOGNITION FOR SIGN LANGUAGE

by

İsmail Arı

B.S, Computer Engineering, Boğaziçi University, 2006

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

Graduate Program in Computer Engineering
Boğaziçi University
2008

FACIAL FEATURE TRACKING AND EXPRESSION RECOGNITION FOR SIGN LANGUAGE

APPROVED BY:

Prof. Lale Akarun (Thesis Supervisor)	....
Prof. Bülent Sankur	....
Assist. Prof. Pınar Yolum	....

DATE OF APPROVAL: 25.07.2008

ACKNOWLEDGEMENTS

With gratitude to my thesis advisor Prof. Lale Akarun for her lasting guidance, support and significant comments throughout the preparation of this thesis.

With gratitude to Prof. Bülent Sankur and Assist. Prof. Pınar Yolum for participating in my thesis jury and giving me valuable feedback.

With gratitude to Prof. Alpar Sevgen for his invaluable contribution to my education and showing me how to enjoy science; to Prof. Cem Ersoy for being a role model for young researchers who want to become a ”multi-professor” successfully balancing between scientific research and extracurricular activities; and to all my teachers who influenced me throughout my education: Enver Çoraplı, Gülsüm Topul, İlknur Direkçi, H. İbrahim Çelik, Prof. Cem Say, Prof. John Freely, Prof. Ethem Alpaydın, Assist. Prof. Burak Acar, Assoc. Prof. Taylan Akdoğan, and Dr. Suzan Üsküdarlı.

With gratitude to Handan Şahin, Ömer Koçağ, Kerem Bölükbaşı, Serbay Zambak, and Güven Demirel with whom I feel very lucky to have met in my life.

With gratitude to Abuzer Yakaryılmaz, Albert Ali Salah, Aslı Uyar, Atay Özgövde, Aydın Ulaş, Berk Gökberk, Elif Sürer, Erinç Dikici, Itır Karaç, İlker Demirkol, Rabun Koşar, and especially to Gökhan Kabukçu for his sincere friendship; Hamdi Dibeklioğlu for accompanying me in the long-lasting nights of thesis runs; Mehmet Gönen for being there when I need any kind of help; Neşe Alyüz for contributing to my cognitive load with her encouragement and her valuable comments on the presentation; Pınar Santemiz for her helps in HMM classification; and Koray Balcı, Onur Dikmen and Oya Aran for sharing their great experiences. I am also grateful to many other CmpE researchers for their kind friendship and valuable discussions.

With gratitude to Abuzer Yakaryılmaz, Amaç Güvensan, Aslı Uyar, Didem Çınar, Hakan Haberdar, İrem Türkmen, Neşe Alyüz, Onur Güngör, Oya Aran, Öner Zafer,

Pınar Santemiz, Rüştü Derici, and Zeyneb Kurt for their efforts in the collection of the video database used in the thesis.

With gratitude to Sinan Ayyıldız and Adem Tosunoğlu from whom I learned more than music. Our lessons inspired me also in scientific research.

With gratitude to my family for their love and support in all stages of my education. I appreciate them for letting me free to take my own steps in life and dedicate this thesis to them.

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

ABSTRACT

FACIAL FEATURE TRACKING AND EXPRESSION RECOGNITION FOR SIGN LANGUAGE

Extracting and tracking facial features in image sequences automatically is a required first step in many applications including expression classification. When sign language recognition is concerned, expressions imply non-manual gestures (head motion and facial expressions) used in that language. In this work, we aimed to classify the most common non-manual gestures in Turkish Sign Language (TSL). This process is done using two consecutive steps: First, automatic facial landmarking is performed based on Multi-resolution Active Shape Models (MRASMs) on faces. The landmarks are fitted in each frame using MRASMs for multiple views of faces, and the best fitted shape which is most similar to the shape found in the preceding frame is chosen. This way, temporal information is used for achieving consistency between consecutive frames. When the found shape is not trusted, deformation of the tracked shape is avoided by leaving that frame as empty and re-initializing the tracker. Afterwards, the empty frames are filled using interpolation, and alpha-trimmed mean filtering is performed on the landmark trajectories to eliminate the erroneous frames. Second, the tracked landmarks are normalized and expression classification is done based on multivariate Continuous Hidden Markov Models (CHMMs). We collected a video database of non-manual signs to experiment the proposed approach. Single view vs. multi-view and person specific vs. generic MRASM trackers are compared both for tracking and expression parts. Multi-view person-specific tracker seems to perform the best. It is shown that the system tracks the landmarks robustly. For expression classification part, proposed CHMM classifier is experimented on different training and test set selections and the results are reported. We see that the classification performances of distinct classes are very high.

ÖZET

YÜZ ÖZNİTELİKLERİNİN TAKİBİ VE İŞARET DİLİ İÇİN İFADE TANIMA

Bir imge dizisinde bulunan yüz öznitelik noktalarının otomatik olarak takip edilmesi, ifade tanımayı da kapsayan birçok uygulamanın ilk adımıdır. İşaret dili özelinde bakarsak, ifadeler hem duygusal ifade hem de baş hareketi içerebilen ele ait olmayan işaretler olarak karşımıza çıkar. Bu çalışmada, Türk İşaret Dili'nde yaygın olarak kullanılan ifadeleri tanımayı amaçladık. Önerdiğimiz sistem iki aşamadan oluşmaktadır: İlkinde, imge dizisindeki her kare için, çok-yönlü (düz, sağa, sola, yukarı) Çok-çözünürlüklü Aktif Şekil Modelleri (ÇÇAŞM) ile yüzdeki nirengi noktaları otomatik olarak saptanır. Bulunan yönlerden şekli modele en iyi oturan ve önceki seçilen şekle en yakın olan yönün şekli seçilir. Eğer seçilen şeklin güvenirliği, eşik değerinin altında ise o kare boş bırakılır ve şekil başlangıç durumuna getirilir. Böylece takip edilen şeklin dağılması önlenir ve sistemin gürbüz çalışması sağlanır. Boş bırakılan kareler interpolasyon ile doldurulur ve hatalı sonuçları elemek için alpha-trim ortalama süzgeci kullanılır. İkinci aşamada takip edilen noktalar normalize edilir ve çok değişkenli Sürekli Saklı Markov Modelleri (SSMM) tabanlı sınıflandırıcıya girdi olarak verilir ve ifade tanınması yapılır. Bulunan sonuçları sınayabilmek için ele ait olmayan ifadelerden oluşan bir video veritabanı topladık. Hem takip hem tanıma kısımları için ÇÇAŞM yöntemini tek-yön/çok-yön ve genel/kişiye-özel çeşitlemeleri ile çalıştırıp sonuçları karşılaştırdık. Çok-yönlü kişiye-özel takipçi en başarılı sonuçları vermektedir ve sistemin gürbüz bir şekilde noktaları takip edebildiği gözlemlenmektedir. Sınıflandırma kısmı için önerilen SSMM sınıflandırıcısını değişik eğitim ve test kümelerinde denedik. Birbirinden farklı sınıflar için başarı çok yüksek gözükmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

a_{ij}	Probability of making a transition from state i to state j
$\mathbf{A}$	State transition matrix of a hidden Markov model
$\mathbf{b}$	Shape parameter vector
$b_j(m)$	Probability of observing the symbol o_m in state j
$\mathbf{b}_g$	Texture parameter vector
$\hat{\mathbf{b}}$	Constrained shape parameter vector
$\hat{\mathbf{b}}_{\mathbf{g}}$	Constrained texture parameter vector
$\mathbf{B}$	Observation matrix of a hidden Markov model
c_t	Normalizing coefficient
c_O	Selected class for observation sequence O
C	Number of distinct non-manual gesture classes
$\mathbf{C}_j$	Covariance of gradients for j^{th} landmark
$\mathbf{C}_g$	Texture covariance matrix
$\mathbf{C}_s$	Shape covariance matrix
dx_{min}	Minimum displacement value in x direction
dx_{max}	Maximum displacement value in x direction
dy_{min}	Minimum displacement value in y direction
dy_{max}	Maximum displacement value in y direction
$\mathbf{e}_k$	k^{th} significant eigenvector
$\mathbf{E}$	Matrix composed of eigenvectors
$\mathbf{E}_c$	Matrix composed of combined eigenvectors
$\mathbf{E}_g$	Matrix composed of texture eigenvectors
$\mathbf{E}_s$	Matrix composed of shape eigenvectors
f	Frame number
F	Total number of frames
F_v	Total number of frames in v^{th} video
$\mathbf{g}$	Normalized texture vector
$\mathbf{g}_{ij}$	Gradient vector of pixel intensities along the profile of p_{ij}
$\mathbf{g}_{im}$	Texture vector before photometric normalization

$\bar{\mathbf{g}}$	Mean texture vector
$\bar{\mathbf{g}}_j$	Mean gradient vector for j^{th} landmark
$\hat{\mathbf{g}}$	Constrained texture vector
$\mathbf{I}$	Identity matrix
l	Current level of multi-resolution pyramid
l_{max}	Top level number of multi-resolution pyramid
L	Number of facial landmarks
m	Search width
m_x	Mean of x values
m_y	Mean of y values
M	Number of distinct observable symbols
n	Length of a texture vector *or* Profile width
N	Number of distinct hidden Markov model states *or* Number of images in the training set
N_{max}	Maximum number of iterations allowed at each level of multi-resolution pyramid
o_m	Distinct observation symbol
O	Observation sequence
O_t	Observation symbol at time t
p_{ij}	j^{th} landmark in the i^{th} shape
$\mathbf{p}_j$	Point
$\mathbf{p'}_j$	Target point
r	Convergence ratio
rms	Root mean square error
$r^{'}$	Accepted convergence ratio
Q	State sequence
$\mathbf{s}_i$	Shape vector
$\mathbf{s}_{ref}$	Reference shape vector
$\bar{\mathbf{s}}$	Mean shape vector
$\hat{\mathbf{s}}$	Constrained shape vector
$\hat{\mathbf{s}}_i$	Aligned shape vector
S_i	State i of a hidden Markov model

t	A triangle in T *or* a time step
t'	Target triangle
T	Delaunay triangulation *or* Sequence length
$\mathbf{v}_i$	Vertex of a triangle
$\mathbf{v_i}^{'}$	Target vertex of a triangle
V	Total number of videos
$\mathbf{V}^i$	i^{th} video
W	Number of distinct views
$\mathbf{W}$	Reliability matrix of landmarks
$\mathbf{W}_s$	Weight matrix used to commensurate shape and texture units
x	x coordinate value
y	y coordinate value

α	Blending parameter *or* scale
α_i	Lagrange coefficient
$\alpha_t(i)$	Forward variable
β	Blending parameter *or* offset
$\beta_t(i)$	Backward variable
γ	Blending parameter
$\gamma_t(i)$	State posterior probability of a hidden Markov model
λ_k	k^{th} significant eigenvalue
μ_{ik}	Mean of the k^{th} Gaussian component in state i
$\mathbf{\Phi_g}$	Sample texture space
$\mathbf{\Phi_s}$	Sample shape space
π_i	Probability of initially being at state i
$\mathbf{\Pi}$	Vector containing the initial state probabilities
Σ_{ik}	Covariance of the k^{th} Gaussian component in state i
Θ	Parameter set
$\xi_t(i,j)$	State transition posterior probability of a hidden Markov model

2D	Two Dimensional

3D	Three Dimensional
AAM	Active Appearance Model
ASM	Active Shape Model
AU	Action Unit
CHMM	Continuous Hidden Markov Model
DHMM	Discrete Hidden Markov Model
CMU DB	Carnegie Mellon University Database
DNMF	Discriminant Non-negative Matrix Factorization
DP	Dynamic Programming
EM	Expectation Maximization
FACS	Facial Action Coding System
HCI	Human-Computer Interaction
HSI	Hue Saturation Intensity
HMM	Hidden Markov Model
IR	Infrared
JAFFE	Japanese Female Facial Expression Database
kNN	k-Nearest Neighbor
LDA	Linear Discriminant Analysis
LNMF	Local Non-negative Matrix Factorization
MPT	Machine Perception Toolbox
MRASM	Multi-resolution Active Shape Model
NMF	Non-negative Matrix Factorization
NN	Neural Networks
OpenCV	Open Source Computer Vision Library
PBVD	Piecewise Bézier Volume Deformation
PCA	Principle Component Analysis
RGB	Red Green Blue
ST	Spatio-temporal
SVM	Support Vector Machine
TSL	Turkish Sign Language

1. INTRODUCTION

There has been a growing interest in the field of extracting and tracking facial features and understanding the expression of the subject especially in the last two decades. The pioneering work of Ekman and Friesen [1] contributed considerably in perspective of psychology discipline and this valuable work initiated the involvement of the computer science community to analyze facial expressions in an automatic way in the following years.

1.1. Motivation

Facial expression recognition is related to many sources in the human as seen in Figure 1.1, which is modified from Fasel and Luettin's survey [2] on automatic facial expression. Sign language is one of these sources.

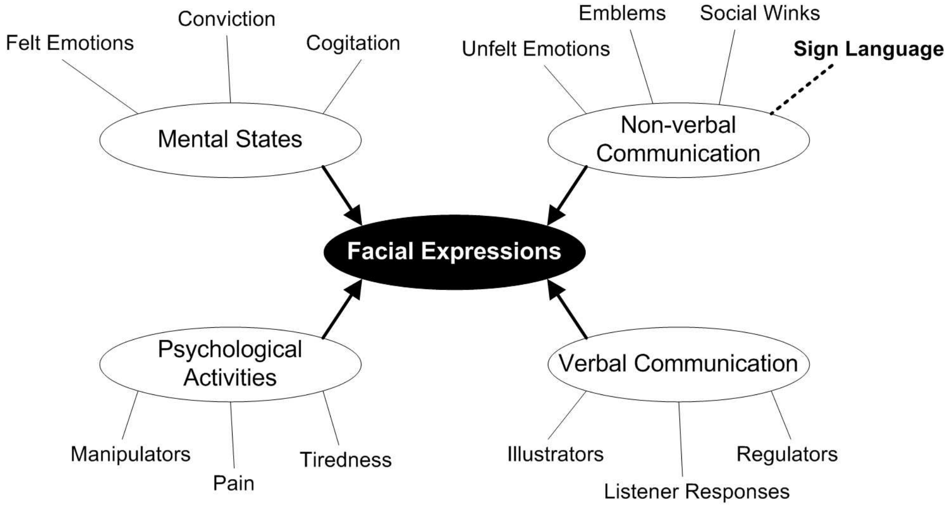


Figure 1.1. Sources of expressions (modified from [2])

One of the major application areas for a robust facial feature tracking and expression classification system is the sign language analysis. Sign language expressions are composed of manual (hand gestures) and non-manual components (facial expressions, head motion, pose and body movements). Some expressions are performed only using hand gestures whereas some facial expressions change the meaning of the accompanying hand gestures. Therefore, a robust high-performance facial feature tracker and facial-expression classifier is a required component in sign language recognition.

There are many other applications where facial feature tracking and expression recognition are needed. An example of such a system is the improvement of driver prudence and accident reduction. The driver's face is tracked while she is driving and she is warned if there seems to be an alerting fact that can result in an accident such as sleepy eyes, yawning too much or looking out of the road.

Furthermore, with a facial feature tracker, it becomes possible to play a synthesized avatar so that it imitates the expressions of the performer. The facial expression of a subject can be synthesized on an avatar in instant messenger applications by only sending facial features instead of sending the video of the face.

Human-Computer Interaction (HCI) systems may also be enriched by a facial feature tracker. For a user who is incapable of using her hands, a facial expression controller may be a solution to send limited commands to a computer.

In this work, we are interested in facial landmark tracking for sign language recognition but the developed techniques may be adapted for other purposes.

1.2. Literature Review

There are tens of researchers dealing with the automatic facial feature tracking and facial expression recognition problem and we examine some remarkable studies to summarize the ongoing scientific research in these areas. Some surveys on the subject are available, such as Fasel and Luettin' review [2] of the ongoing research on automatic facial expression analysis, Ong and Ganganath's survey [3] of automatic sign language analysis which mainly focuses on manual signs used in sign languages, and Pantie and Rothkrantz's work [4] that examines the state of the art approaches in automatic analysis of facial expressions.

Different approaches have been tried in facial expression analysis systems. Most approaches include three distinct phases: First, before a facial expression can be analyzed, the face must be detected in a scene. This process is followed by the extraction of the facial expression information from the video and localizing (in static images) or tracking (in image sequences) these features under different poses, illumination, ethnicity, age and expression. The outputs of this process are given as the input for the following stage, which is the recognition of the expression. This final step is a classification stage where the expression is classified into one of the predefined classes of expressions.

1.2.1. Face Detection and Tracking

Face detection is the first stage which is desired to be automated. In most of the research, face is already cropped and the analysis starts with feature extraction and tracking. In the rest, automated face detectors are used. These can be classified mainly into two classes: vision-based detection and detection using infrared (IR) cameras.

Spors and Rabenstein [5] use skin color detection and principal component analysis (PCA) based eye localization to locate the face for their tracking algorithm. To reduce the computational complexity further, the eye detection and tracking task is divided into two steps in their work. First the eye is localized. When the position of the eyes is known, tracking is performed using a luminance-adapted block matching technique.

Jordao *et al.* [6] and Ahlberg [7] use skin color detection followed by biggest blob selection in their work.

On the other hand, Kapoor and Picard [8] prefer to use IR cameras for pupil localization in their head nod and shake detection system. Similarly, Zhang and Ji's initializer [9] is based on IR cameras. An IR camera takes two images of the scene at each frame where the pupils reflect in one of the pairs and all the other objects remain the same. So, a simple subtraction of two images gives the location of the pupils.

There are also free face detection software available to researchers for usage and improvement. Most popular of these is the face detector of Open Source Computer Vision Library (OpenCV), [10] which depends on Haar-like wavelet-based object detection proposed by Viola and Jones in [11]. Another available software is the Machine Perception Toolbox [12] which also works similarly.

The face detection systems discussed are summarized in Table 1.1.

Table 1.1. Face Detection Techniques

Vision Based Detection	Skin Color + PCA [5]
	Skin Color + Biggest Blob [6, 7]
	Haar-like Features [10, 12]
Detection Using IR Cameras	IR Pupil Tracking [8, 9]

1.2.2. Facial Feature Extraction and Tracking

Numerous features have been applied to the facial expression recognition problem. Image-based models rely on the pixel values of the whole image (holistic) or related parts of the image (local). On the other hand, model-based approaches create a model that best represents the face by using training images. Feature points are also used as features to feed in the classifier or to play an avatar. Difference images are used to find the eye coordinates from the image pairs gathered by IR cameras. In the initial research done in this area, markers were used to analyze the facial data. In addition, optical flow and motion models are also used in feature extraction and tracking. A categorization of the related work is given in Table 1.2 to draw a mental map of the approaches used. We describe each approach briefly:

It is seen that image-based and model-based approaches are more dominant in the literature. As an image-based technique, Gabor wavelets are widely used in facial feature detection. Dubuisson *et al.* [14] apply triangulation to the magnitude of the filtered image which is passed through the Gabor kernel. Then, they detect the three boxes containing the facial features (eye regions and the mouth region) with a

Table 1.2. Categories of Facial Feature Extraction Techniques

	Holistic Methods	**Local Methods**
Image-based	Gabor filters [13]	Local Gabor filters [14]
	Non-negative matrix factorization (NMF) / Local NMF (LNMF) [16]	High gradient components [15]
		Grayscale morphological filters [6]
	Principal components [13, 17]	Neural networks (pixel values) [18]
Model-based	AAM [7, 19, 20]	
	ASM /Active contours [21, 22]	
	3D Deformable models with optical flow [23]	
	3D Deformable Models / PBVD [24, 25]	
Dense Optical Flow		Region-based flow [15]
Motion Models		Block matching [26]
Feature Point Tracking		Feature point tracking [9, 15]
Difference Images	Holistic difference-images [8]	
Marker-based		Makeup/highlighted features [27]

classification of the regions laying in the convex envelope of the triangulation.

Gokturk *et al.* [23] create a 3D deformable model from stereo tracking and apply PCA in their study. The resulting model approximates any generic shape as a linear combination of shape basis vectors. The additional optical flow tracking computes the translational displacement of every point.

Sebe *et al.* [24] use piecewise Bézier volume deformation (PBVD) tracker which is developed by Tao and Huang [28]. This face tracker uses a model-based approach where an explicit 3D wireframe model of the face is constructed. Cohen [25] also uses this tracker to recognize facial expressions from video sequences using time information in his thesis.

A novel approach is to use a cropped region and to apply neural network classification to the pixels within the cropped region as Franco and Treves [18] use in their work.

Furrows play an important role in differentiating the expressions, so furrow detection is another valuable feature as Lien e*t al.* [15] showed in their work. This approach is mainly based on high gradient component detectors which work horizontally. They also use dense flow sequences and feature points in the upper face.

The results of the research done by Calder *et al.* [29] give us an intuition about the functionality of the principal component axes that come out from the sample expression set when they examine the dominating variance axes with Principal Component Analysis (PCA).

Statistical approaches have three main stages: “capture”, “normalization” and “statistical analysis”. In brief, in the capture part, one defines a certain number of points (landmarks) on the contour of the object in question for shape and uses image warping for texture. The following shape normalization is done using Procrustes Analysis and texture normalization is done by removing global illumination effects between frames. Finally, Principal Components Analysis (PCA) is performed to analyze the variances between object shapes or textures and this information is also used for synthesis. Active Shape Models (ASMs) and Active Appearance Models (AAMs) are two widely used statistical approaches where both of them are proposed by Cootes *et al.* in [21] and [19] respectively.

The AAM approach is used in facial feature tracking due to its ability in detecting the desired features as the warped texture in each iteration of an AAM search approaches to the fitted image. Ahlberg [7], and Abboud and Davoine [20] use AAM in their work.

In addition, ASMs - which are the former version of the AAMs that only use shape information and the intensity values along the profiles perpendicular to the shape surface are also used to extract features such as the work done by Votsis *et al.* [22].

Some chosen methods in facial feature extraction are given in Table 1.3.

1.2.3. Facial Expression Recognition

Although some works focus on feature extraction and do not involve classification [5–7, 23, 26], the ideal facial expression analyzer needs a classifier. Various machine learning algorithms are applied to the extracted facial features in this manner.

Kapoor and Picard [8] use Hidden Markov Models (HMMs) to classify the pupil trajectories into one of the two classes they define which are head nodding and shaking. Lien *et al.* [15] also prefer HMMs for classification of the extracted features to one of the three action units (AUs) they choose. Their feature extraction method depends on facial feature point tracking, dense flow tracking, and high gradient component detection. Similarly, HMMs are used by Cohen [25] in his thesis to classify the facial expressions performed by five subjects into one of the six universal expression classes. These classes are happiness, anger, surprise, disgust, fear, sadness as discussed by

Table 1.3. Some Facial Feature Extraction Techniques

Author(s)	**Method**	**Comment**
Calder *et al.* [29]	PCA and LDA	Gives important components to identify expressions
Cootes *et al.* [19]	Feature Points after AAM Fit	Costly
Jordao *et al.* [6]	Color-based Face detection and Morphological filters to find eyes, nose and mouth	Heuristic Approach
Ahlberg [7]	AAM fitting, following FACS chosen (1) jaw drop (2) lip stretcher (3) lip corner depressor (4) upper lip raiser (5) eyebrow lowerer (6) outer eyebrow raiser	Encoded for MPEG-4
Lien *et al.* [15]	Three approaches: feature tracking, dense optical flow, furrow detection and tracking	Upper face only
Franco and Treves [18]	Pixel values to feed in Neural Network	Needs a standard region of a face
Buciu and Pitas [16]	PCA, NMF and LNMF	The non-negative constraints are imposed to be consistent with the neurophysiological fact that the neural firing rate is non-negative.

Ekman and Friesen [1]. It is seen that HMMs are frequently chosen for classification because of their good results in gesture (temporal data) recognition.

Zhang and Ji [9] use Dynamic Bayesian Networks in their research to find the performed AU among 18 different AUs, where they rely on the feature points tracked in image sequences.

Sebe *et al.* [24] experiment with different types of classifiers such as k-Nearest Neighbor (kNN), Support Vector Machines (SVMs), Bayesian Networks and decision-tree based classifiers in their work: Authentic Facial Expression Analysis. The outstanding technique is the kNN, as they report.

In the work of Franco and Treves [18], a rectangular region of the face that involves one eye and half of the mouth and nose is cropped from the images. The pixel values of this cropped rectangle are given as input to the neural networks (NN) classifier for classification into one of neutral, happy, sad or surprised expressions.

In addition to analysis from image sequences, there is also work done on still images. Buciu [30] applies discriminant non-negative matrix factorization (DNMF), Abboud and Davoine apply decision tree based classifier [20], and Buciu and Pitas [16] apply nearest neighbor using cosine similarity measure and maximum correlation classifier to the images selected from Cohn-Kanade image database [31]. Dubuisson *et al.* [14] also use the same database and perform two types of classification. A binary classifier is used to distinguish between two confusing classes and a 6-class classifier is used for general classification,

A brief summary and categorization of the classification approaches and their results can be seen in Table 1.4.

Table 1.4. Classification Methods and Results

Author(s)	**Method**	**# classes**	**Extraction Methods**	**Test Cases**	**Accuracy**
Analysis from static images					
Buciu [30]	Discriminant NMF (DNMF)		no info	234 CMU DB images [31]	82.85% (max)
Dubuisson [17]	Decision-Tree Based Classifier	6	PCA, LDA	345 samples	85.8%
Franco and Treves [18]	Neural Networks	4 (neutral, happy, sad, surprise)	Clipping a region	14 subjects	84.5%
Abboud and Davoine [20]	no info about classification algorithm	6+1(neutral)	Compared asymmetric bilinear factorization to LDA	108 test images 70 training images CMU DB [31]	no info
Buciu and Pitas [16]	Nearest neighbor using cosine similarity measure and maximum correlation classifier	6+1(neutral)	no info	213 images in (JAFFE) [32] 234 images in CMU DB [31]	81%
Dubuisson *et al.* [14]	Binary classifier (1) 6-expression classifier (2)	6	PCA, LDA	120 images in FERET DB [33] 550 images in CMU DB [31]	88%-92% (1) 68%-75% (2)
Analysis from facial image sequences					
Kapoor and Picard [8]	HMM for pupil trajectories	2 (head nod, head shake)	IR pupil tracking	110 sample sequences	78.46%
Sebe *et al.* [24]	SVM, Bayesian networks, Decision-tree based classifier, kNN	no classes		Own DB (db1) and CMU DB (db2) [31]	Err. in pxl: 4.43 (db1) 6.96 (db2)
Cohen [25]	DP time alignment (1), emotion specific HMM (2), multi-level HMM (3)	6	PCA	5 subjects	52%(1), 55%(2), 58%(3)
Lien *et al.* [15]	HMM	3 AUs	Feature point tracking (1), dense flow tracking (2), high gradient component detection (3)	$>$ 260 image sequences and 5000 images	85% (1,3) 93% (2)
Zhang and Ji [9]	Dynamic Bayesian Networks	6	Feature point tracking	-	$<$ 3 pxls in points ˜97% in classes

1.2.4. A Desired Expression Recognition System

We explored the three main stages related to facial expression analysis in the previous subsections. But starting from a primitive tracker, a facial expression analyzer system should have numerous capabilities to perform well on different conditions. In other words, any age, ethnicity or outlook variance should be handled by the system. In addition, the system should be robust in the presence of different illumination conditions and partially occluded faces. Although special markers can make the analysis of facial feature variations easier, a desired system should track faces without makeup or markers. Subjects may appear with different poses to the camera or may change their angles during the expressions, so, rigid head motions should be dealt with. Automation of face detection, facial feature extraction and expression classification are vital. During the acquisition, there may be misleading or missing data, therefore inaccurate facial expression data is also a problem. Classifying Facial Action Codes (FACS) would also play a significant role since Ekman and Friesen [1] clarified their importance in universal facial expressions. Finally, the desired system should run in real time. All these properties of a targeted recognizer (modified from the survey of Pantie and Rothkrantz [4]) are summarized in Table 1.5.

After discussing the related work of the researchers, we give a comparison of the referenced works in Table 1.6.

1.2.5. Facial Expression Classification in the Scope of Sign Language Expressions

Sign language expressions are performed with the combination of manual (hand gestures) and non-manual components (facial expressions, head motion and pose, body movements). Some expressions are performed only using hand gestures whereas some change the meaning where a facial expression accompanies hand gestures. For example, in Turkish Sign Language (TSL), a sentence can be in positive, negative and question clause forms if the hand gesture of the verb is accompanied by different non-manual gestures as Zeshan [35] describes in her work on TSL. Thus; when we refer to *sign*

Table 1.5. Properties of a Desired Recognizer

1	Subjects of any age, ethnicity and outlook
2	Deals with variation in lightning
3	Deals with partially occluded faces
4	No special markers/makeup required
5	Deals with rigid head motions
6	Automatic face detection
7	Automatic facial feature extraction
8	Deals with inaccurate facial expression data
9	Automatic facial expression classification
10	# interpretation categories
11	Classifying facial action codes
12	# facial action codes
13	Runs on real-time

Table 1.6. Comparison of Recognizers in the Scope of Properties in Table 1.5

Author(s)	**Characteristics of an ideal automated facial expression analyzer**												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Sobottka and Pitas [26]	x	x	x	o	-	x	o	x	-	-	-	-	x
Gokturk *et al.* [23]	x	x	x	o	o	o	o	x	-	-	-	-	x
Spors and Rabenstein [5]	x	x	x	o	-	o	-	-	-	-	-	-	o
Kapoor and Picard [8]	x	x	x	o	-	x	pupils	x	o	2	x	x	o
Jordao *et al.* [6]	x	x	x	o	o	o	o	x	-	-	-	-	near
Ahlberg [7]	x	x	x	o	o	o	o	x	-	-	-	-	near
Sebe *et al.* [24]	-	-	x	o	o	x	o	x	o	4	x	x	x
Lien *et al.* [15]	o	o	x	o	-	x	o	x	x	x	o	3	x
Cohen [25]	x	x	x	o	o	x	o	x	o	6	o	12	x
Franco and Treves [18]	-	o	x	o	x	x	o	x	o	4	x	x	x
Zhang and Ji [9]	-	-	o	o	$< 30^o$	o	o	o	o	6	o	18	o
Dubuisson *et al.* [14]	-	-	-	o	x	x	o	x	o	6	x	x	x
Ari *et al.* [34]	x	x	x	o	o	o	o	x	o	7	x	x	x

Legend: 'o' means yes, 'x' means no, '-' means not available or not known *language expressions*, we do imply that that they are composed of facial expressions (e.g. happiness, sadness, anger) and/or head movements (e.g. head shaking, nodding). In some parts of the text, the terms *non-manual expression* and *facial sign* are used in a similar meaning.

Aran *et al.* [36] select their video database such that it involves some signs where the manual gestures are the same but different facial expressions and head movements occur.

Therefore, a robust high-performance facial feature tracker and facial-expression classifier is a must in sign language recognition.

Most of the work done about sign language recognition focuses on hand gesture recognition and lacks non-manual sign analysis. Ma and Gao [37] use only hand gestures in their work on Chinese Sign Language classification. Similarly, Bowden *et al.* [38] and Sagawa and Takeuchi [39] take only hand gestures into consideration in their sign language interpreter and recognizer.

There are also systems that only require features to synthesize the signs using an avatar. In [40], Zhao *et al.* use hand positions to visualize the words in their English to American Sign Language translation system.

As stated, most of the research focuses on classifying expressions into one of Ekman and Friesen's [1] six universal expressions: happiness, anger, surprise, disgust, fear, sadness. But in the scope of sign language, all of these expressions would not be needed and smaller number of classes would be sufficient as Zeshan [35] states for TSL where the robustness and speed of the system is very important.

1.3. Approach and Contributions

In [41], Keskin et al. proposed a Hidden Markov Model based real time hand tracker that works with colored gloves. This system is capable of reconstructing 3D locations of the hand and 3D gesture recognition was established in real time.

Motivated from the previous work, Aran *et al.* [42] enhanced the gesture recognizer such that it can classify the performed manual gesture into one of the seven signs chosen from Turkish Sign Language (TSL). The user watches pre-recorded signs via the user interface and tries to perform the selected sign. Interactive learning of the signs is achieved by this method.

After identifying the need of non-manual gestures to classify a sign language word correctly, Aran *et al.* integrated a facial expression and head movement tracker and classifier in [36]. In this work, we chose 19 expressions from American Sign Language where some expressions are the same in hand gestures but differ in head motions or facial expressions. The purpose of the system was to distinguish these type of signs from each other as well as identifying the performed sign. The classification of hand gestures was near accurate but the non-manual gesture classification was working poorly. Thus, it led to a deficiency in the the sign classification, especially in distinguishing between similar classes.

We required a video database of facial signs on which proposed algorithms could be tested in order to develop a distinct system that can deal with facial signs. The databases open to researchers in the literature mostly involve static images. There exist image sequences involving facial expressions but they lack head movements performed with facial expression. To overcome this need, we introduced a video database of non-manual signs selected from TSL in [43]. An appearance analysis of the facial signs existing in the database and their classification from manually annotated landmarks are given in this work as sample research showing how to use the database.

The contribution of this thesis falls in two parts which we will describe throughout the thesis. First, we develop a multi-view facial landmark tracker that acts robustly in image sequences. This tracker is an extension of Multi-resolution Active Shape Model approach of Cootes *et al.* [21]. Secondly, we propose a facial sign recognizer that interprets the facial landmark locations throughout the image sequence which depends on a Continuous Hidden Markov Model classifier. The flowchart of the proposed system can be seen in Figure 1.2.

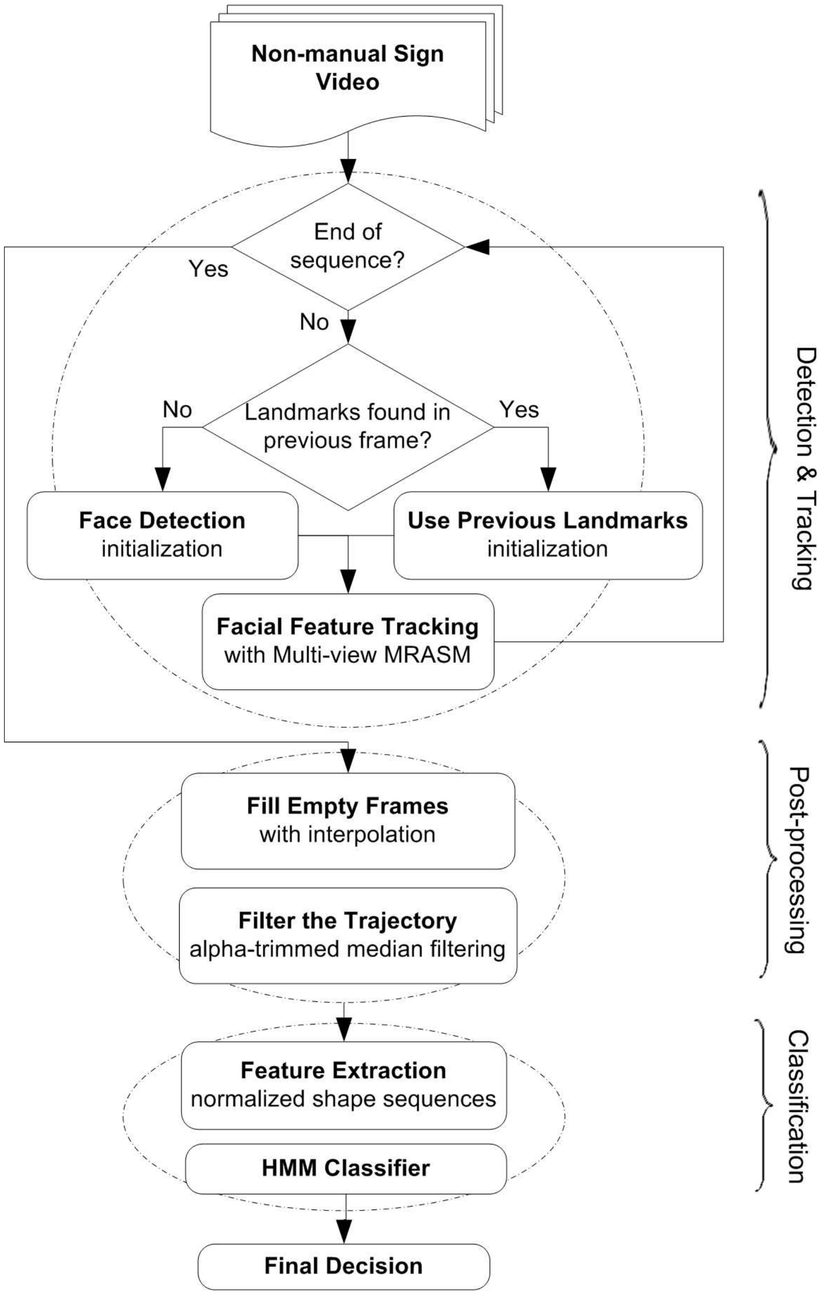

Non-manual Sign Video
End of sequence?
Yes
No
Landmarks found in previous frame?
No
Yes
Face Detection initialization
Use Previous Landmarks initialization
Facial Feature Tracking with Multi-view MRASM
Fill Empty Frames with interpolation
Filter the Trajectory alpha-trimmed median filtering
Feature Extraction normalized shape sequences
HMM Classifier
Final Decision
Detection & Tracking
Post-processing
Classification

Figure 1.2. Flowchart of the proposed system

1.4. Outline of the Thesis

In this chapter, an introduction is made by describing the problem which is the recognition of facial expressions and head movements (i.e. facial signs) and the motivation to solve this problem. Then, a detailed literature survey followed by our approach and contributions is made.

Chapter 2 describes the mathematical details of the facial feature tracking in sign videos. It starts with describing statistical analysis of shapes and appearance and continues with the construction of Active Shape Models, and their extension to work in multi-view and multi-resolution. The details of fitting a shape to an unseen image and tracking these landmarks throughout the image sequence conclude this chapter.

In Chapter 3, the mathematical details of classifying the tracked facial landmarks using Continuous Hidden Markov Models is explained. This chapter gives a background for Discrete and Continuous Hidden Markov Models and then explains the normalization procedure of the facial landmark locations before giving as an input to the HMM classifier.

Chapter 4 includes the experiments tested on a video database which is composed of non-manual signs and presents the achieved results . Both tracking and classification experiments are involved in this chapter.

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

2. FACIAL FEATURE TRACKING

2.1. A Review of Statistical Approaches for Shape Analysis and Tracking

Cootes *et al.* [21] introduced Active Shape Models (ASMs) which are used to fit a shape to an unseen image where the shape deformation is done in two steps: First, the landmarks are translated along the profiles perpendicular to landmark locations where the new location gives the minimum gradient error between the fitted profile and the mean profile that is calculated by averaging the profile gradients of all shapes in training set. Secondly, the shape deformation is projected onto a new shape space where the projected shape is similar to those in the training set. This way, the shape is safely deformed and the irrelevant deformation of the shape is avoided. In [44], Cootes and Taylor describe multi-resolution approach to ASMs which enhances the performance of fitting both in time complexity and in accuracy. In Multi-resolution ASM (MRASM), the image is downsampled into smaller dyadic sizes of the original size and first fit is performed in the smaller sample. After each level, the fitted shape is used as initial shape in the larger sample and the ASM search is done.

In [19], Cootes *et al.* introduced a new active model named as Active Appearance Model (AAM). In AAMs, the intensity values of pixels in the convex hull of the landmark locations are taken into consideration instead of relying only on the landmark profiles. A uniform warping followed by global illumination removal is performed to normalize the objects and Principle Component Analysis (PCA) is done using these samples. The fitting of the shape to an object is done by iteratively perturbing the appearance such that each synthesized object of an iteration is closer to the target image. AAMs are explored in detail by Stegmann [45] in his thesis where he shows their usage in medical applications as well as facial feature fitting and tracking. He also introduces an AAM toolbox in [46] open for researchers to experiment with AAMs.

There is a considerable difference between ASMs and AAMs in terms of time complexity. In ASMs, only the pixels along the landmark profiles are used whereas all the pixels in the convex hull of the shape are used in AAMs. Additionally, warping the convex hull to get equivalent number of pixels in each iteration is a time consuming operation.

The applicability of these approaches to object segmentation and tracking led researchers to investigate ASMs and AAMs in detail. An extension was the 3D active models. Temporal information was taken into account and earlier 2D spatial models were extended in order to use motion information through time. Hamarneh and Gustavsson [47] introduced spatio-temporal shapes (ST-shapes) which are the 2D+time extensions of ASMs. In [48], Mitchell *et al.* show the segmentation of volumetric cardiac magnetic resonance (MR) images and echocardiographic temporal image sequences using 3D AAM training and fitting.

Furthermore, appearance models considering the color information in RGB, YUV and HSI color spaces were examined by Koschan *et al.* [49].

In this chapter, we explore 2D MRASMs and show how to extend them to work in multiple views of faces and use them for tracking facial landmarks.

2.2. Statistical Analysis of Shapes

The statistical analysis of shapes consists of three distinct steps which are capture, normalization and PCA (Principle Component Analysis) as shown in Figure 2.1.

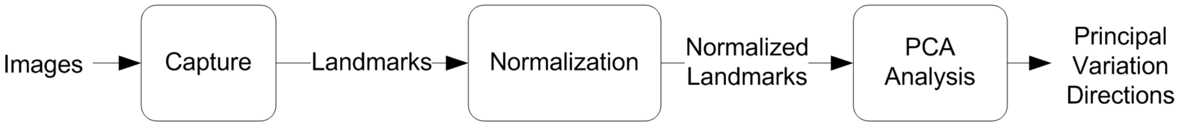


Figure 2.1. The phases in statistical analysis of shapes

2.2.1. Capturing Landmarks

We first gather a sample set of N images where each image involves a face and then we decide on L feature points (landmarks) which are common in all faces in this set such as the points shown in Figure 2.2.

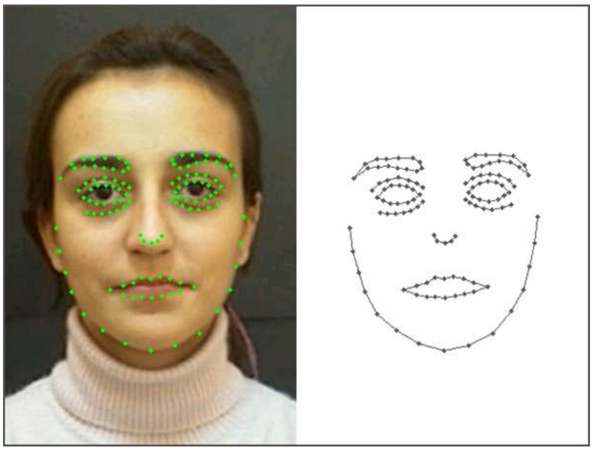


Figure 2.2. Selected feature points

We manually annotate these L selected landmarks in each image and create the sample shape space $\mathbf{\Phi_s}$ containing shapes $\mathbf{s}_i$ where

$$\mathbf{s}_i = (x_1, y_1, x_2, y_2, \ldots, x_L, y_L)\,, \quad i = 1, \ldots, N$$

is a shape containing the coordinates of the landmarks in that image.

2.2.2. Shape Normalization

We want to model the variability using PCA which is an eigen-analysis of shape dispersions in $2L$-dimensional space. Applying PCA to non-normalized landmark locations would lead to unexpected results. Since shape definition should be independent of similarity transformations (translation, scaling and rotation), one can see that shape alignment is crucial to overcome PCA defects. The normalization is done using Procrustes Analysis which has the following steps as Cootes and Taylor describe in [44]:

1 - Choose the first shape as the reference (estimate of the mean) shape;

2 - Align all remaining shapes to it;

3 - Recalculate the mean shape from all;

4 - Repeat steps 2-3 until the mean shape converges.

Figure 2.3. Shape Alignment Algorithm

In step 2 in Figure 2.3, the aligned shape is translated, scaled and rotated to best match the reference shape. To align a shape $\mathbf{s}_i$ in $\mathbf{\Phi_s}$ to the reference shape $\mathbf{s}_{ref}$, $\mathbf{s}_i$ is mapped to $\hat{\mathbf{s}}_i$ such that the distance between $\hat{\mathbf{s}}_i$ and $\mathbf{s}_{ref}$ is minimized. Let the distance be,

$$d^2_{ref,\hat{i}} = \left(\hat{\mathbf{s}}_i - \mathbf{s}_{ref}\right)^T \mathbf{W}^T\mathbf{W} \left(\hat{\mathbf{s}}_i - \mathbf{s}_{ref}\right) \tag{2.1}$$

where $\mathbf{W}$ is chosen to be a diagonal matrix that involves the reliability values of landmarks with a mean value of 1.

For the j^{th} landmark in $\mathbf{s}_i$, the following similarity transformation is used

$$\begin{bmatrix} \hat{x}_{ij} \\ \hat{y}_{ij} \end{bmatrix} = \begin{bmatrix} scos(\theta) & -ssin(\theta) \\ ssin(\theta) & scos(\theta) \end{bmatrix} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{2.2}$$

Eq. 2.2 can be rewritten as follows:

$$\begin{bmatrix} \hat{x}_{ij} \\ \hat{y}_{ij} \end{bmatrix} = \begin{bmatrix} x_{ij} & -y_{ij} & 1 & 0 \\ y_{ij} & x_{ij} & 0 & 1 \end{bmatrix} \begin{bmatrix} scos(\theta) \\ ssin(\theta) \\ t_x \\ t_y \end{bmatrix} \tag{2.3}$$

Then, this equation can be written for all points (by adding rows to the formula above for other landmarks) and the following form can be achieved:

$$\mathbf{s}_i = \mathbf{A}\mathbf{z} \tag{2.4}$$

where the dimensions of $\mathbf{s}_i$, $\mathbf{A}$ and $\mathbf{z}$ are $2L \times 1$, $2L \times 4$ and 4×1 respectively.

Since there will not be a unique solution due to the fact that the number of observations is more than the number of unknowns, least squares approach is applied by integrating Eq. 2.4 into Eq. 2.1, and solve to find $\mathbf{z}$ and the similarity parameters needed to align $\mathbf{s}_i$ to $\mathbf{s}_{ref}$.

$$\mathbf{z} = (\mathbf{A}^T\mathbf{W}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^T\mathbf{W}\mathbf{s}_{ref} \tag{2.5}$$

If $\mathbf{W}$ is taken as identity matrix $\mathbf{I}$ for simplicity, Eq. 2.5 becomes

$$\mathbf{z} = \mathbf{A}^{+}\mathbf{s}_{ref} \tag{2.6}$$

where $\mathbf{A}^{+}$ is the pseudo-inverse of $\mathbf{A}$.

Alignment of a shape onto the reference shape is illustrated in Figure 2.4.

When faces in sign language videos are considered, rotation carries vital information that we would like to retain in the model, so that its removal is excluded in shape normalization. Thus, the parameters are found by removing the rotation information from Eq. 2.2 and deriving the rest of the equations in a similar way. A sample alignment of facial landmarks is shown in Figure 2.5.

2.2.3. Modelling the Shape Variance with Principal Component Analysis

There exists a redundancy stemming from inter-point correlations in the shape space as Stegmann states in his thesis work [45]. A classical statistical method for

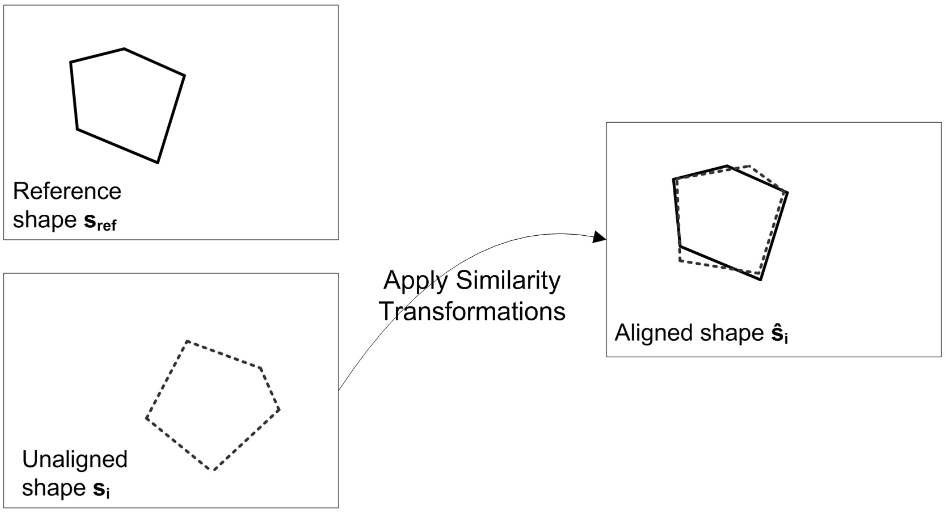

Reference shape $\mathbf{s}_{\mathbf{ref}}$
Unaligned shape $\mathbf{s}_{\mathbf{i}}$
Apply Similarity Transformations
Aligned shape $\hat{\mathbf{s}}_{\mathbf{i}}$

Figure 2.4. Alignment of $\mathbf{s}_i$ on $\mathbf{s}_{ref}$

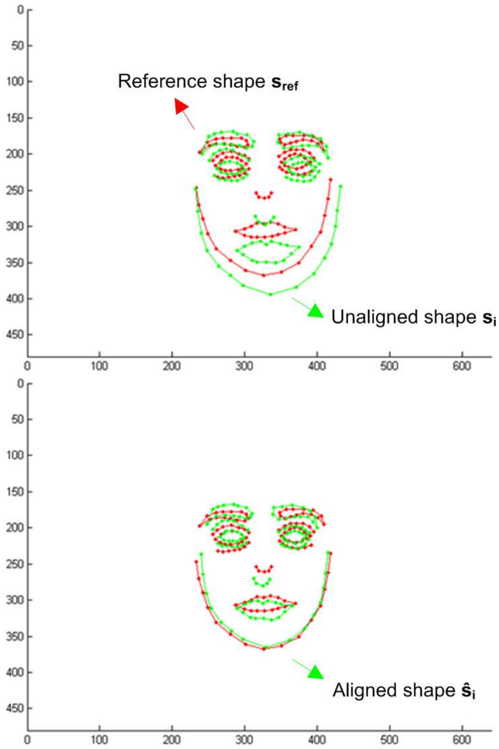


Figure 2.5. Alignment of facial landmarks in $\mathbf{s}_i$ on $\mathbf{s}_{ref}$

eliminating such redundancy is the Principal Component Analysis (PCA) which is invented by Pearson [50] and based on his work, introduced in [51] by Harold Hotelling. PCA is a vector space transform often used to reduce multidimensional data sets to lower dimensions for analysis as described in detail in the Principle Component Analysis book of Joliffe [52]. Depending on the field of application, it is also named the discrete Karhunen-Loève transform, the Hotelling transform or proper orthogonal decomposition (POD).

The main motivation in using PCA is the dimensionality reduction where most of the total variance (e.g. 95%) can be represented using a small number of orthogonal basis vectors. But, conversely, PCA will enable us to synthesize new shapes which are similar to those in the sample space. Indeed, this will be very useful when we need to re-synthesize the fitted shape to a new one which is more reasonable (i.e. more similar to the training shapes) and avoid modeling non-relevant shape variations.

Applying PCA to the normalized shape space $\mathbf{\Phi_s}$ gives us the major principles containing the most variation in the space in question. In other words, PCA can be seen as an eigen-analysis of normalized shape space. The PCA steps applied to the normalized shapes are given in the algorithm in Figure 2.6.

$$\bar{\mathbf{s}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_i \tag{2.7}$$

$$\mathbf{Q} = \left[\begin{array}{cccc} \mathbf{s}_1 - \bar{\mathbf{s}} & \mathbf{s}_2 - \bar{\mathbf{s}} & \cdots & \mathbf{s}_N - \bar{\mathbf{s}} \end{array} \right]_{2L\times N} \tag{2.8}$$

$$\mathbf{C}_s = \frac{1}{N}\mathbf{Q}^T\mathbf{Q} \tag{2.9}$$

$$\mathbf{C}_s\mathbf{e}_k = \lambda_k\mathbf{e}_k \tag{2.10}$$

```
**Require** All shapes $\mathbf{s_i},\ i = 1, 2, \ldots, N$ are normalized
Compute the mean shape $\bar{\mathbf{s}}$ using Eq. 2.7;
Form the $N \times 2L$ matrix $\mathbf{Q}$ as defined in Eq. 2.8;
**if** $N < 2 \times L$ **then**
  $\mathbf{Q} \Leftarrow \mathbf{Q}^T$ ;
**end if**
Compute the covariance matrix $\mathbf{C}_s$ using Eq. 2.9;
Decompose $\mathbf{C}_s$ to its eigenvectors $\mathbf{e}_k$ and eigenvalues $\lambda_k$ satisfying Eq. 2.10;
**if** $N < 2 \times L$ **then**
  **for** $k = 1$ to $K$ **do**
    $\mathbf{e}_k \Leftarrow \mathbf{Q}\mathbf{e}_k$ ;
    $\mathbf{e}_k \Leftarrow \mathbf{e}_k / ||\mathbf{e}_k||$ (normalization);
  **end for**
**end if**
```

Figure 2.6. Principal Component Analysis Algorithm

Any shape $\mathbf{s}$ can also be described using all eigenvectors in a lossless way and the coefficients of eigenvectors form the parameter vector $\mathbf{b}$.

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{E}_r \mathbf{b} \tag{2.11}$$

$$\mathbf{b} = \mathbf{E}_r^T (\mathbf{s} - \bar{\mathbf{s}}) \tag{2.12}$$

where

$$\mathbf{E}_r = \begin{bmatrix} \mathbf{e}_1 & \dots & \mathbf{e}_r \end{bmatrix}, \quad r = \min(2L, N)$$

As stated earlier, the motivation behind applying PCA is to reduce the dimension and use $K < r$ eigenvectors, yet preserving the most of the variation. K is chosen where it satisfies

$$\sum_{k=1}^{K} \lambda_k \geq 0.95 \times \sum_{k=1}^{r} \lambda_k$$

Let $\lambda_1, \dots, \lambda_k$ be the first K eigenvectors. Then using $\hat{\mathbf{b}} = (b_1, \dots, b_K)$, we can synthesize $\hat{\mathbf{s}}$ which is an estimate of $\mathbf{s}$ that is similar to the shapes in $\mathbf{\Phi}_{\mathbf{s}}$.

$$\hat{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{E}_K \hat{\mathbf{b}} \tag{2.13}$$

$$\hat{\mathbf{b}} = \mathbf{E}_K^T (\mathbf{s} - \bar{\mathbf{s}}) \tag{2.14}$$

2.3. Statistical Analysis of Appearance

To synthesize a complete image of an object or structure, we must model both its shape and its texture (the pattern of intensity or color within the object region) as Cootes and Taylor [44] states. The phases of appearance analysis can be seen in Figure 2.7.

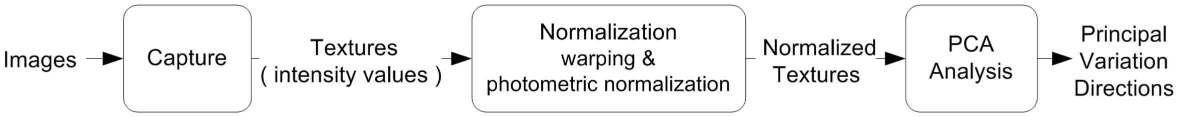


Figure 2.7. The phases in statistical analysis of textures

2.3.1. Gathering Intensity Values

Let a *face texture* be the intensity values (grayscale or colored) of pixels that reside in the convex hull of the face shape. So, the face textures are gathered when the landmarks are annotated in each face image. Different face textures can be seen on the left of Figure 2.8. Let $\mathbf{\Phi_g}$ be the space of sample training textures.

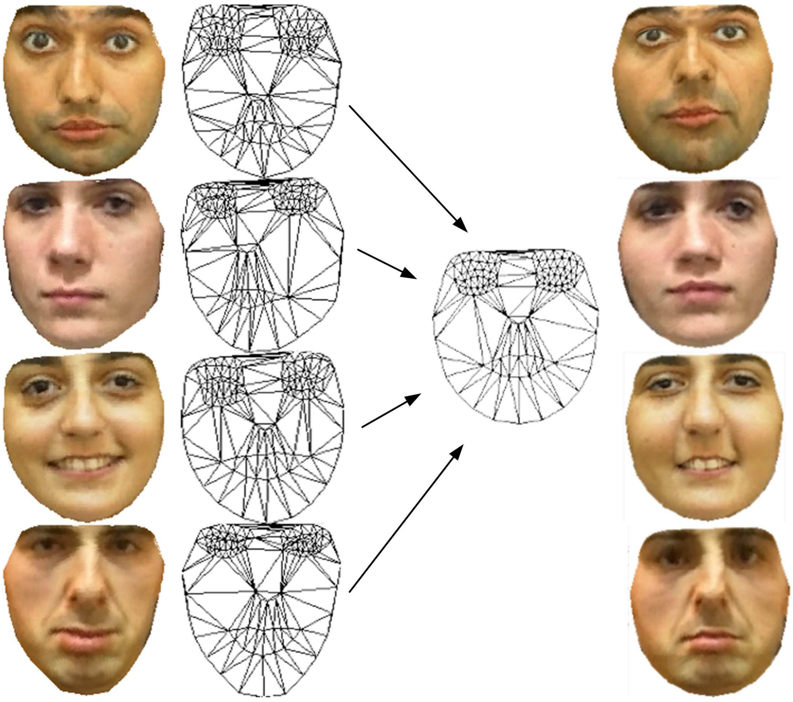


Figure 2.8. Normalization of face textures by triangulation matching

2.3.2. Texture Normalization

2.3.2.1. Texture Warping by Triangulation. To build a statistical model of texture variations in the training set, the number of pixels should be the same for each sample texture vector that is taken into consideration. In order to satisfy this, each face image is warped to the mean shape by using Delaunay triangulation as seen in Figure 2.8. Let T be the Delaunay triangulation of the mean shape $\bar{\mathbf{s}}$ and t be a triangle of T as illustrated in Figure 2.9.

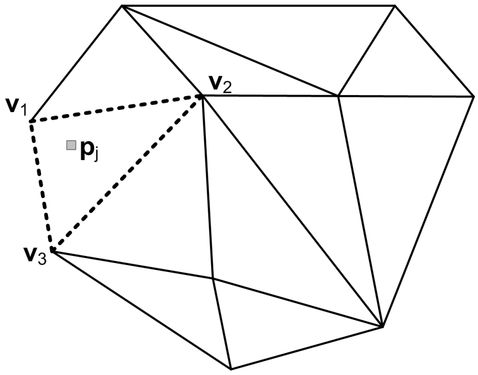


Figure 2.9. Delaunay triangulation of a shape

And let t' be the target triangle for t as shown in Figure 2.10.

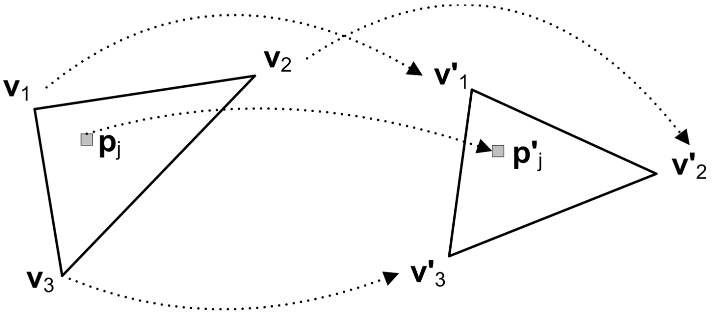


Figure 2.10. Point mapping in triangles

So, we can write

$$\alpha \mathbf{v}_1 + \beta \mathbf{v}_2 + \gamma \mathbf{v}_3 = \mathbf{p}_j \tag{2.15}$$

$$\alpha + \beta + \gamma = 1 \tag{2.16}$$

where $\mathbf{p}_j$ involves the coordinates of the point, and $\mathbf{v}_k,\ k = 1, 2, 3$ are the vertices of the triangle.

Then, the detailed steps of the warping algorithm are constructed as in Figure 2.11.

```
Choose $\bar{\mathbf{s}}$ as reference shape $\mathbf{s_{ref}}$ and triangulate it using Delaunay triagulation $T$;
for each triangle $t \in T$ do
  for each pixel $j \in t$ do
    Find the blending values $\alpha, \beta$ and $\gamma$ satisfying Eq. 2.15 and Eq. 2.16.;
    Save the corresponding blending parameters;
  end for
end for
for each texture $i$ in $\mathbf{\Phi_g}$,   $i = 1, \ldots, N$ do
  Use the same triangulation $T$ to triangulate this shape;
  for each pixel $j$ in $\mathbf{s_{ref}}$ do
    Find the triangle it belongs to in shape $i$, thus the three vertices which will be necessary for describing the interior points;
    Retrieve the stored blending parameters belonging to this point, insert them to Eq. 2.15 with the vertices found in the previous part;
    Find the coordinates of intensity value which is mapped to coordinate $j$ (Bilinear interpolation may be used [45]);
    Set the new pixel value;
  end for
end for
```

Figure 2.11. Texture Warping Algorithm

As a result of the texture alignment (warping), the dimension (number) of texture pixels is equalized in all samples.

2.3.2.2. Photometric Normalization. To minimize the effect of global lightning variation, each texture $\mathbf{g}_{im}$ in $\mathbf{\Phi_g}$ is normalized by applying a scale α and an offset β as described in [44],

$$\mathbf{g} = (\mathbf{g}_{im} - \beta\mathbf{1})/\alpha \tag{2.17}$$

Let $\bar{\mathbf{g}}$ be the mean of the normalized $\mathbf{\Phi_g}$ so that the sum of elements of $\mathbf{g}$ is zero and the variance is unity. Then, α and β are chosen as

$$\alpha = \mathbf{g}_{im}.\bar{\mathbf{g}}, \qquad \beta = \mathbf{g}_{im}.\mathbf{1}/n \tag{2.18}$$

where n is the number of elements in $\mathbf{g}_{im}$. In practice, $\bar{\mathbf{g}}$ is initialized as the mean texture vector. Then, the normalization process is done by normalizing the textures and re-estimating the mean from them for a few iterations. This step is called the removal of global illumination.

2.3.3. Modelling the Texture Variance with Principal Component Analysis

Similar to the modelling of shape variance previously, we can model the texture variation using PCA. Remember that n is the number of elements in any texture vector, then we construct the covariance matrix as follows

$$\bar{\mathbf{g}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{g}_i \tag{2.19}$$

$$\mathbf{Q}_\mathbf{g} = \left[\begin{array}{cccc} \mathbf{g}_1 - \bar{\mathbf{g}} & \mathbf{g}_2 - \bar{\mathbf{g}} & \cdots & \mathbf{g}_N - \bar{\mathbf{g}} \end{array}\right]_{n\times N} \tag{2.20}$$

$$\mathbf{C}_g = \frac{1}{N}\mathbf{Q}_g^T\mathbf{Q}_g \tag{2.21}$$

and decompose it to its eigenvalues as follows:

$$\mathbf{C}_g\mathbf{e}_k = \lambda_k\mathbf{e}_k \tag{2.22}$$

The PCA algorithm described in Figure 2.6 holds for texture analysis where the length of the vectors is n instead of $2L$. The alternative calculation included in the algorithm may decrease calculation time when there are fewer samples than vector length because we have generally $n \gg N$ in practice [45].

Any texture $\mathbf{g}$ can be described using all eigenvectors in a lossless way and the coefficients of eigenvectors form the parameter vector $\mathbf{b}_g$.

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{E}_r\mathbf{b}_g \tag{2.23}$$

$$\mathbf{b}_g = \mathbf{E}_r^T(\mathbf{g} - \bar{\mathbf{g}}) \tag{2.24}$$

where

$$\mathbf{E}_r = \begin{bmatrix} \mathbf{e}_1 & \dots & \mathbf{e}_r \end{bmatrix}, \quad r = \min(n, N)$$

As in shape analysis, k most significant eigenvectors describing most of the total variation can be retained. Then using $\hat{\mathbf{b}}_g = (b_1, \dots, b_K)$, we can synthesize $\hat{\mathbf{g}}$ with fewer parameters which is an estimate of $\mathbf{g}$.

$$\hat{\mathbf{g}} = \bar{\mathbf{g}} + \mathbf{E}_K\hat{\mathbf{b}}_g \tag{2.25}$$

$$\hat{\mathbf{b}}_g = \mathbf{E}_K^T(\mathbf{g} - \bar{\mathbf{g}}) \tag{2.26}$$

2.3.4. Modelling Both Shape and Appearance Variance

It was seen that a sample can be reconstructed using model parameters of shape $\mathbf{b}_s$, and texture $\mathbf{b}_g$. To remove the correlation between shape and texture parameters and to make the model more compact, an additional PCA is performed on the concatenated parameters $\mathbf{b}$ which can be written as

$$\mathbf{b} = \begin{pmatrix} \mathrm{W}_s\mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathrm{W}_s\mathbf{E}_s^T(\mathbf{s} - \bar{\mathbf{s}}) \\ \mathbf{E}_g^T(\mathbf{g} - \bar{\mathbf{g}}) \end{pmatrix} \tag{2.27}$$

where the subscripts s and g in $\mathbf{E}$ imply shape and texture eigenvectors respectively. $\mathbf{W}_s$ is used to make the units in shape (point locations) and texture (intensity values) commensurate and is calculated as $\mathbf{W}_s = r\mathbf{I}$ where r^2 is the ratio of total intensity variation to the total shape variation.

A further PCA is applied to $\mathbf{b}$ vectors to give

$$\mathbf{b} = \mathbf{E}_c\mathbf{c} \tag{2.28}$$

where $\mathbf{E}_c$ are the eigenvectors and $\mathbf{c}$ holds the appearance parameters that controls both shape and texture in a single vector. From the linear nature of the model, it is possible to get shape and texture data as follows:

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{E}_s\mathbf{W}_s^{-1}\mathbf{E}_{cs}\mathbf{c} \tag{2.29}$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{E}_g\mathbf{E}_{cg}\mathbf{c} \tag{2.30}$$

where

$$\mathbf{E}_c = \begin{pmatrix} \mathbf{E}_{cs} \\ \mathbf{E}_{cg} \end{pmatrix}$$

2.4. Constructing the Active Shape Model

Let p_{ij} be the j^{th} landmark in the i^{th} shape, such that $p_{ij} = (x_{ij}, y_{ij})$. $\mathbf{g}_{ij}$ is the gradient of pixel intensities along the profile of p_{ij} as in Figure 2.12. The gradient is calculated by taking the difference of consecutive intensities. A more complicated shape and the profiles of its landmarks are given in Figure 2.13.

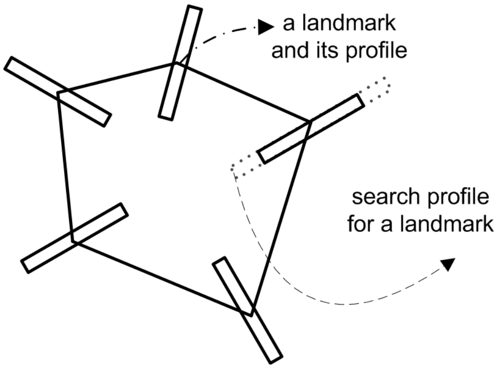


Figure 2.12. Profiles of landmarks in a shape

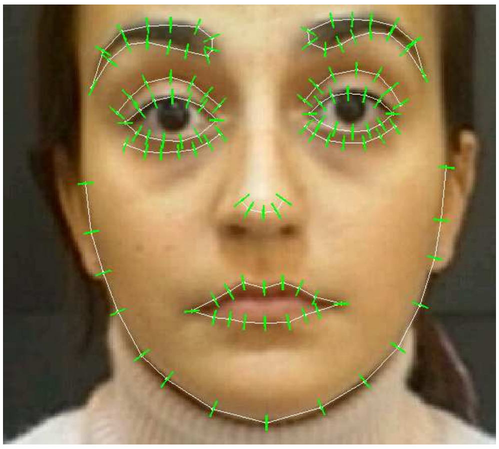


Figure 2.13. Profiles of landmarks in a a more complicated shape

To illustrate, the intensity values of the red channel on the profiles of lips in Figure 2.13 and the corresponding gradient values are given in Figure 2.14. The grayscale image on the left shows the intensity values for each point profile in each row. The bar graphs in each row on the right corresponds to the intensity change (gradient values) for that point. Since there are 16 landmarks on lips, Figure 2.14 includes 16 rows. It is seen that nearly all of the points are selected where there is considerable change in the intensity values (i.e. the points are selected on edges).

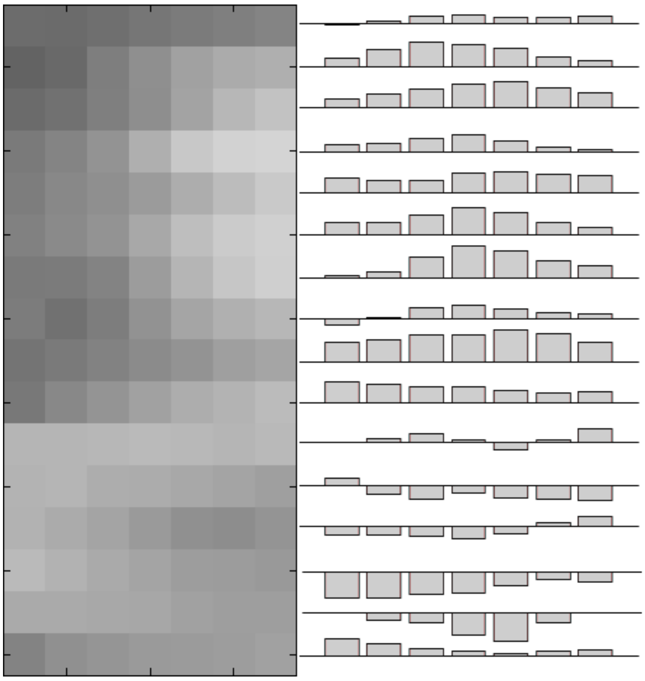


Figure 2.14. Intensities along the profiles around lips with the corresponding gradients

Then we calculate $\bar{\mathbf{g}}_j$ as the mean gradient vector and $\mathbf{C}_j$ as the covariance of gradients for each landmark. Thus a single Active Shape Model (ASM) is composed of particular $\bar{\mathbf{s}}$, $\mathbf{E}_K$, λ_k, $\bar{\mathbf{g}}_j$ and $\mathbf{C}_j$ ($k = 1, \ldots, K$ and $j = 1, \ldots, L$).

2.5. Fitting a Shape to a Test Image

The initialization is done by detecting the face using OpenCV's face detector [10] and $\bar{\mathbf{s}}$ is placed on the found face. Then, the shape is iteratively perturbed along the profile until convergence. Each iteration involves two steps as follows:

2.5.1. Finding the Best Fit

Let us say n is the profile width (length of the model mean vector $\bar{\mathbf{g}}_j$) and m is the search width (length of the sampled profile gradient vector) as in Figure 2.12 where $m > n$.

For each landmark, we find the best fit along the profile where the best profile gradient $\hat{\mathbf{g}}_j$ gives the minimum Mahalanobis distance with the model, i.e. the term

$$(\hat{\mathbf{g}}_j - \bar{\mathbf{g}}_j)^T \mathbf{C}_j^{-1} (\hat{\mathbf{g}}_j - \bar{\mathbf{g}}_j) \tag{2.31}$$

is minimized.

For example, let the sampled profile have gradient values as shown in Figure 2.15. If the model has mean value $\bar{\mathbf{g}}_j$ as given in the figure, the best fit will be 2 units far along the profile to the current landmark location as seen as gray colored bar in the lowest plot of Figure 2.15. Remember that $\mathbf{C}_j$ is also required to calculate the cost of fit but it is not shown in the figure.

It is also seen from Figure 2.15 that there are a total of $m - n + 1$ candidate locations for best fit along each profile.

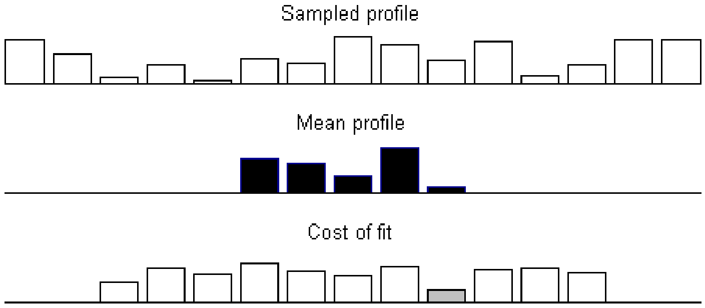


Figure 2.15. Search along the sampled profile and the best fit location

2.5.2. Constraining the Best Fit

The best fit is constrained by finding the approximate shape parameters $\hat{\mathbf{b}}$ using Eq. 2.26 and constraining each coefficient b_k satisfying $-3\sqrt{\lambda_k} \geq b_k \leq 3\sqrt{\lambda_k}$ for $k = 1, \ldots, K$. That is, if the value is out of the allowed limits, then it is changed to the nearest allowed value. This way, the fitted shape avoids deformation and will be similar to the ones in $\mathbf{\Phi_s}$. In Figure 2.16, it is clear that the deformation of the shape is avoided where the best fit (on the left) and the corresponding constrained shape (on the right) in an ASM iteration are shown.

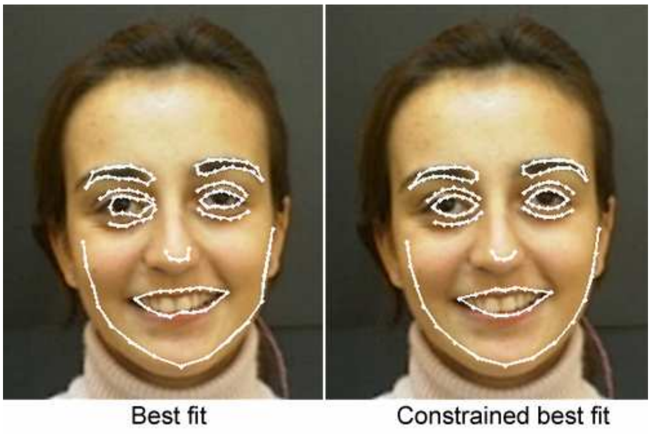


Figure 2.16. Best fit is constrained by projecting $\mathbf{b}$ back to $\hat{\mathbf{b}}$ using Eq. 2.26.

2.6. Multi-resolution Approach

In the multi-resolution approach, instead of using a single level ASM search, a model is created for each level of the image pyramid where the original size images are in the lowest level and higher models involve sub-sampled images. The search is first done at the highest level and the found shape is passed to the lower level as the initial shape for that level. So a rough estimate is found with less computational cost in the highest level and fine-tuned at each level it goes through. This procedure is called Multi-resolution ASM (MRASM).

Let l be the current level of the multi-resolution pyramid, l_{max} be the top level number, N_{max} be the maximum number of iterations allowed at each level, r be the convergence ratio that shows the ratio of landmarks staying unchanged after an iteration (proportion of points found within $(n-1)/4$ units of current position) and $r^{'}$ the accepted convergence ratio that is sufficient to stop the search. The full MRASM search algorithm becomes as given in Figure 2.17.

```
$l \Leftarrow l_{max}$ (i.e. choose top level) ;
Initialize the shape ;
while $l \geq 0$ do
  if $l \neq l_{max}$ then
    Scale the shape coordinates by 2 for consistency at this new level ;
  end if
  (asm-1) Find the best fit as explained in Subsection 2.5.1 ;
  (asm-2) Constrain the best fit as explained in Subsection 2.5.2 ;
  if not ($r > r^{'}$ or $N_{max}$ iterations are performed in this level ) then
    return to step (asm-1) in this loop ;
  end if
  $l \Leftarrow l - 1$ ;
end while
```

Figure 2.17. Multi-resolution ASM Search Algorithm

2.7. Data Refitting

Since $\mathbf{\Phi_s}$ is gathered by clicking feature points manually, the training set is error-prone to human mistakes. To reduce this bias, data refitting is performed. So, a model is trained using $\mathbf{\Phi_s}$, which involves the shapes gathered manually. Then, for each image in the training set, MRASM search is performed but the shape is initialized by using the ground truth shape instead of initializing with the mean shape. Thus each shape is refitted using this model. Finally, a new model is trained by using the fitted shapes as described in the work of Gross et al. [53].

2.8. View-based ASM Modelling and Searching

Cootes *et al.* [54] extended AAMs by training view-based models in their work. The main motivation here is to group the samples with the same view (gaze or pose) in the same set and interpret the facial expression variations with fewer eigenvectors instead of interpreting the head pose change variations. Since there is head motion in addition to facial expressions in sign videos, a single view model is not sufficient for handling all views of the face and we extend ASMs to work in multiple views. When the subject changes the head pose, some components of the face such as eyebrows and eyes have considerably different gradient changes on the contours as seen in Figure 2.18. For example, the end of the eyebrows is completely occluded in the rightmost image and the intensity change in the eye region that is near to nose varies among different views.

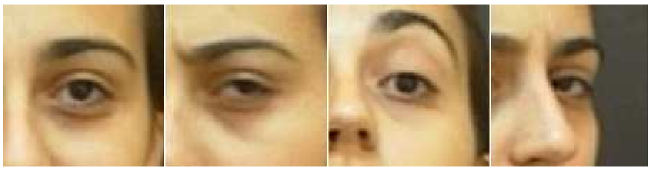


Figure 2.18. Left eye and eyebrow in different views

So, the training set is divided into W subsets and a different model is trained for each view where $W = 4$ and the views v_i $(i = 1, 2, 3, 4)$ are frontal, left, right and upwards views in our case. Sample images are shown in Figure 2.19.

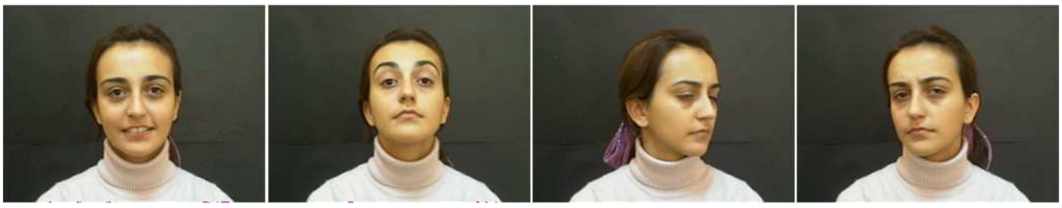


Figure 2.19. The four selected views

The search in an image is done using each model and the best fitting model is selected such that it gives the minimum root mean square error with the model. That is, root mean square error is similarity metric for us to decide on a model view.

Formally, if $\bar{\mathbf{g}}_{v_i,p}$ is the concatenated vector of the mean profile gradients $\bar{\mathbf{g}}_j$, $j = 1, \ldots, L$ of view v_i, and $\mathbf{g}_{v_i,p}$ is the concatenated vector of found profile gradients using this model, then

$$v_{best} = \arg\min_{v_i} E(\mathbf{g}_{v_i,p}, \bar{\mathbf{g}}_{v_i,p}) \tag{2.32}$$

where $E(\mathbf{x}, \mathbf{y})$ is the root mean square error between $\mathbf{x}$ and $\mathbf{y}$ calculated as

$$E(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{\sum_{i}^{n}(x_i - y_i)^2}{n}} \tag{2.33}$$

Let us simply denote $E(\mathbf{g}_{v_i,p}, \bar{\mathbf{g}}_{v_i,p})$ as rms_{g,v_i} for further usage.

2.9. Tracking in an Image Sequence

In a still image where there is no temporal information, we simply selected the model giving the minimum error. When time is also introduced and we have image sequences instead of still images, we may want the tracked shape to be consistent through the frames. Although we find the best fit, the fitted shape may not be an acceptable result. So there are two important rules as follows:

1. If the search results are not acceptable, leave the frame as empty.
2. If there are many accepted search results, choose the one that most resembles the accepted shape in the preceding frame.

Let $rms_{t,v_i,f}$ be the root mean square error between shapes in the f^{th} frame found with the v_i^{th} model and the immediate preceding accepted frame, where $f = 2, \ldots, F$. If the previous frame is already accepted as a valid fit, then the most previously accepted frame number is $f - 1$. This metric informs us about the change in shape in the given interval and is calculated by modifying Eq. 2.33 for the shape vectors in question.

We assume that the first frame is frontal, so we start with a single view MRASM search for $f = 1$. Then the algorithm is given in Figure 2.20.

```
Initialize the face shape
Apply MRASM search for frontal view and fit the face shape where $f = 1$ ;
for $f = 2$ to $F$ do
   Set the previously found shape as initial shape ;
   for each view $v_i$ (frontal, left, right, upwards) do
      Apply MRASM search with the corresponding model ;
   end for
   Eliminate the models whose $rms_{g,v_i}$ is above a threshold ;
   if  no model remains then
      Mark this frame as empty (not found) ;
   else
      Accept the shape fitted by the model giving the minimum $rms_{t,v_i,f}$ ;
   end if
end for
```

Figure 2.20. Algorithm of Tracking Landmarks in Image Sequence

The threshold used in Figure 2.20 can be automatically generated from the fitted shape in the first frame since the subject is assumed to start in neutral face and the MRASM search is relied on to fit the face. The threshold selection is important in tracking. The higher the threshold value, the more frames are rejected and left empty. So, the tracking remains stable and can not catch changes. On the other hand, it the threshold value is selected low, fewer searches will be rejected and thus, the shape can diverge far from the actual face shape.

2.10. Postprocessing the Found Landmarks

2.10.1. Interpolation of Empty Frames

Some frames are left empty for the untrusted MRASM searches during the tracking algorithm. These empty frames are filled by interpolation. For any empty frame f_j with found frames f_i and f_k where $i < j < k$, its shape $\mathbf{s}_j$ is found as

$$\mathbf{s}_j = \frac{(f_k - f_j)\mathbf{s}_i + (f_j - f_i)\mathbf{s}_k}{f_k - f_i} \tag{2.34}$$

2.10.2. Filtering the Landmarks Trajectories

In some of the frames, the tracker is error-prone to spiky changes. So, α-trimmed mean filter is applied to eliminate the spikes encountered during tracking. In addition, it smoothes the trajectories. Suppose that each landmark trajectory throughout time is put in a vector forming a temporal signal. Then, a window of length 5 is traversed through this temporal vector, the lowest and the highest values are excluded and the mean of the remaining three values is taken as the filtered value in our case. In Figure 2.21, a signal including a spiky peak and valley errors is given with the filtered values. It is clearly seen that the spikes are eliminated and the signal is smoothed.

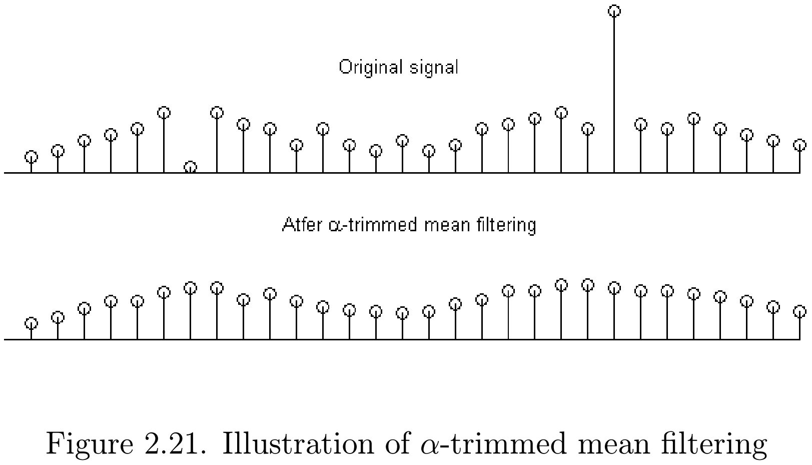


Figure 2.21. Illustration of α-trimmed mean filtering

3. EXPRESSION RECOGNITION

In the previous chapter, we have tracked L facial landmarks in each frame of a sign video. As described in Chapter 1, an ideal system should also classify these landmark motions to any of the expression classes in question. Remember that *expression* is used for non-manual/facial signs in our case, and it does not imply merely a facial expression; it may also involve head pose changes (head movements).

For classifying the performed non-manual sign, Hidden Markov Models (HMMs) are used with the normalized landmark locations as features. We start with summarizing our prior expression classification work based on Support Vector Machines. Afterwards, the details of the feature extraction and the HMM classification investigated in this thesis are given.

3.1. Expression Recognition based on Support Vector Machines

In the joint work with Aslı Uyar [34], we explored expression recognition that consists of two sub-stages which are motion feature extraction and classification using Support Vector Machines (SVMs). The extracted coordinates of facial landmarks in consecutive frames of the video sequences are used to evaluate the maximum displacement values for each feature point in four directions x_+, x_-, y_+ and y_- across the entire image sequence.

3.1.1. Motion Feature Extraction

Displacement based and time independent motion feature vector is used as the input to the SVM classifier. The motion feature vector includes information about both the magnitude and the direction of motion for each landmark. We find the maximum displacement of landmarks where peak location for each landmark may be in different frames.

Let $\mathbf{V}^i$ be the i^{th} video composed of consecutively tracked shapes as follows

$$\mathbf{V}^i = \left[\mathbf{s}_1^i, \ \ \mathbf{s}_2^i, \ \ \cdots \ \ \mathbf{s}_F^i\right] \tag{3.1}$$

where

$$\mathbf{s}_f^i = \left(x_f^{i,1} \ \ y_f^{i,1} \ \ x_f^{i,2} \ \ y_f^{i,2} \ \ \cdots \ \ x_f^{i,L} \ \ y_f^{i,L}\right) \tag{3.2}$$

is the set of tracked landmarks in the f^{th} frame of i^{th} video. Note that, the notation of indices for a point coordinate is modified to enable showing video, frame and landmark number in a single notation.

For each video, the initial frame (i.e. $\mathbf{s}_1^i$) is chosen as the reference frame and the displacements of the landmarks between each frame and the reference frame have been measured.

Then, the maximum displacement values of each landmark in four directions have been chosen as the motion features.

$$\begin{aligned}
dx_{max}^{i,l} &= \max_f \left\{x_f^{i,l} - x_1^{i,l}\right\} \\
dx_{min}^{i,l} &= \min_f \left\{x_f^{i,l} - x_1^{i,l}\right\} \\
dy_{max}^{i,l} &= \max_f \left\{y_f^{i,l} - y_1^{i,l}\right\} \\
dy_{min}^{i,l} &= \min_f \left\{y_f^{i,l} - y_1^{i,l}\right\}
\end{aligned}$$

The output of this process is a single motion vector $\mathbf{z}^i$ for each video.

$$\mathbf{z}^i = \left(dx_{max}^{i,1} \cdots \ dx_{max}^{i,L} \ dx_{min}^{i,1} \ \cdots \ dx_{min}^{i,L} \ \ dy_{max}^{i,1} \ \cdots \ dy_{max}^{i,L} \ dy_{min}^{i,1} \ \cdots \ dy_{min}^{i,L}\right) \tag{3.3}$$

3.1.2. Classification Using SVM

Because of the superior classification performance and its ability to deal with high dimensional input data, SVM was chosen as the classifier in this prior study for facial expression recognition. A brief definition of SVM is given below:

Given a set of training data pairs (x_i, y_i), $y_i \in \{+1, -1\}$, the aim of the SVM classifier is to estimate a decision function by constructing the optimal separating hyperplane in the feature space [55]. The key idea of SVM is to map the original input space into a higher dimensional feature space in order to achieve a linear solution. This mapping is done using kernel functions. Final decision function is in the form:

$$f(x) = \left(\sum_i \alpha_i y_i K(x_i \cdot x) + b \right) \tag{3.4}$$

where $K(x_i \cdot x)$ is the Kernel transformation. The training samples whose Lagrange coefficients α_i are non-zero are called *support vectors* (SV) and the decision function is defined by only these vectors.

3.2. Expression Recognition based on Hidden Markov Models

In this study, we explore the classification of non-manual signs that is composed of two stages: normalization of tracked landmarks in the image sequences and classification based on Hidden Markov Models (HMMs).

3.2.1. Feature Extraction: Normalization of Tracked Shape Sequences

The tracked landmarks found in videos are inconsistent because the scales and positions of the faces in different videos vary. For example, the subject in the i^{th} video may be nearer to the camera, whereas j^{th} video involves a subject performing the sign far from the camera. In order to achieve consistency, a normalization of shape sequences given in Eq. 3.1 should be performed as described in Figure 3.1. Briefly, the first shape of a video is translated to the origin and scaled to unity with a transformation and this transformation is applied to all remaining frames in the same video.

```
for each video $\mathbf{V}^i$ do
    Compute the mean of the first shape $\mathbf{s}_1^i$ as follows ;
        $m_x = \left(\sum_l x_1^{i,l}\right)/L \;, \quad m_y = \left(\sum_l y_1^{i,l}\right)/L$
    Compute the Frobenius Form [45] as follows ;
        $r = \sqrt{\sum_l (x_1^{i,l} - m_x)^2 + (y_1^{i,l} - m_y)^2}$
    Find the transformation $\mathbf{T}$ that translates $\mathbf{s}_1^i$ to the origin and the scaling $\mathbf{S}$ that scales it to unit shape and the unified transformation matrix $\mathbf{A}$;
        $\mathbf{A} = \mathbf{ST} \;, \text{ where } \mathbf{T} = \begin{bmatrix} 1 & 0 & \text{-m}_x \\ 0 & 1 & \text{-m}_y \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{S} = \begin{bmatrix} 1/\text{r} & 0 & 0 \\ 0 & 1/\text{r} & 0 \\ 0 & 0 & 1 \end{bmatrix}$
    for $f = 1$ to $F$ do
        Transform $\mathbf{s}_f^i$ to $\mathbf{s}_f^{\prime i}$ using $\mathbf{A}$ ;
    end for
    Form the normalized video $\mathbf{V}^{\prime i}$ from the transformed shapes ;
end for
```

Figure 3.1. Algorithm of Shape Sequence Normalization

3.2.2. Classification Using Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical model in which the modeled system is assumed to be a Markov process with unknown parameters, and the objective is to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example for classification of sequential data.

In [56], Rabiner emphasizes the outstanding reasons why hidden Markov modeling has become increasingly popular in the last decades: First, the models are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications. Secondly, the models, when applied properly, work very well in practice for several important applications. Alpaydin [57] states that the HMM is a mature technology and HMMs are applied to various sequence recognition tasks such as commercial speech recognition systems in actual use (Rabiner and Juang [58], Jelinek [59]), real-time manual gesture recognition systems (Keskin *et al.* [41], Aran *et al.* [42]) and non-manual gesture recognition systems (Kapoor and Picard [8]).

3.2.2.1. Discrete Markov Processes . Let a system have N distinct states forming $\mathbf{S} = \{S_1, \ldots, S_N\}$. At each time step t, the system is assumed to be in one of the states, that is q_t, where $q_t \in \{s_1, s_2, \ldots, s_N\}$. We assume that only the current state determines the probability distribution for the next state (first order Markovian property) and the transition probability a_{ij} from S_i to S_j under this assumption can be given as:

$$a_{ij} \equiv P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N \tag{3.5}$$

satisfying

$$a_{ij} \geq 0 \text{ and } \sum_{j=1}^{N} a_{ij} = 1 \tag{3.6}$$

The transition probabilities form matrix $\mathbf{A} = [a_{ij}]$ which is an $N \times N$ matrix. Notice that, a_{ij} values show the inner transitions and the initial probabilities are not given yet. We define π_i as the probability that the sequence starts with S_i and $\mathbf{\Pi} = [\pi_i]$

is a vector composed of initial probabilities satisfying

$$\sum_{i=1}^{N} \pi_i = 1 \tag{3.7}$$

3.2.2.2. Hidden Markov Models. Let M be the number of distinct observations which form the set $\mathbf{O} = \{o_1, o_2, \ldots, o_M\}$. In an observable Markov model, $\mathbf{S} \equiv \mathbf{O}$, i.e. we observe the states and we get an observation sequence that is a sequence of states. But in a Hidden Markov Model (HMM), the states are not observable and we observe o_m when the system is in S_j with a probability of

$$b_j(m) \equiv P(O_t = o_m | q_t = S_j) \tag{3.8}$$

where O_t is the observation we get at time t.

N and M are implicitly defined in other parameters, so the model is defined as $\Theta = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$. In HMMs, there are three main problems as follows:

1. **The Evaluation Problem:** Given a model Θ and a sequence of observations $O = \{O_1 O_2 \ldots O_T\}$, what is the probability that the observations are generated by the model, $P(O|\Theta)$?
2. **The Decoding Problem:** Given a model Θ and a sequence of observations $O = \{O_1 O_2 \ldots O_T\}$, what is the most likely state sequence in the model that produced the observations?
3. **The Learning Problem:** Given a model Θ and a sequence of observations $O = \{O_1 O_2 \ldots O_T\}$, how should we adjust the model parameters $(\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$, in order to maximize $P(O|\Theta)$?

In expression recognition, we need to solve the first and the third problems which can be called also as *recognition* and *training* using HMMs.

3.2.2.3. The Evaluation Problem. For a sequence classification problem, one is interested in evaluating the probability of any given observation sequence, $O = \{O_1 O_2 ... O_T\}$, given an HMM model, Θ. We can evaluate $P(O|\Theta)$ by trying all possible state sequences as:

$$P(O|\Theta) = \sum_{\text{all possible } Q} P(O, Q|\Theta) \tag{3.9}$$

However, this is not practical because there are N^T possible state sequences [57]. There is an efficient way of solving this problem based on the idea of dividing the observation sequence into two parts: the first one lasting from time 1 until time t, and the second one from time $t+1$ until time T. This approach is called the *forward-backward procedure.*

The probability, or the likelihood, $P(O|\Theta)$ can be calculated in terms of only the forward variable as follows:

$$P(O|\Theta) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.10}$$

where T is the end of sequence and $\alpha_T(i)$ is the forward variable, that is is the probability of observing the sequence $\{O_1 \dots O_T\}$ and being in state i at time T, given the model Θ. The forward variable can be recursively calculated by going forward in time:

$$\alpha_1(j) = \pi_j b_j(O_1) \tag{3.11}$$

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \right] b_j(O_t) \tag{3.12}$$

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

$$P(O|\Theta) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \tag{3.13}$$

where the backward variable, $\beta_t(i)$ is defined as the probability of observing the partial sequence $\{O_{t+1} \dots O_T\}$ given that we are in state i at time t and the model is Θ. The backward variable can be recursively computed by going backwards:

$$\beta_T(i) = 1 \tag{3.14}$$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(O_{t+1}) \tag{3.15}$$

The recursion steps of calculating forward and backward variables given in Eq. 3.12 and Eq. 3.15 are illustrated in Figure 3.2a and Figure 3.2b, recursively [57].

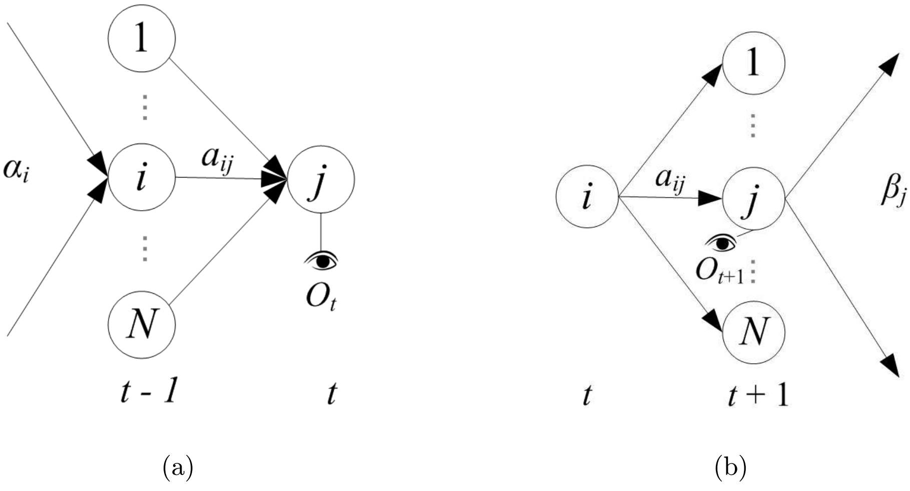


Figure 3.2. Computation of forward variable, $\alpha_t(j)$ and backward variable, $\beta_t(i)$.

When T is large, the computation of the forward variable will lead to an underflow when we want to implement this procedure because recursively multiplying small probabilities will exceed the precision range of the machine. To avoid this, a normalization is performed at each time step. The normalizing coefficient, c_t is calculated as follows:

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)} \tag{3.16}$$

$$\hat{\alpha}_t(i) = c_t \alpha_t(i) \tag{3.17}$$

We also normalize $\beta_t(i)$ values similarly. The computation of $P(O|\Theta)$ must be modified conveniently since $\hat{\alpha}_t(i)$ and $\hat{\beta}_t(i)$ values are already scaled. $P(O|\Theta)$ can be calculated via the normalizing coefficients. However, due to the precision problem, we can only implement this procedure by calculating the logarithm of P as Rabiner explains in [56]:

$$log(P(O|\Theta)) \quad = \quad -\sum_{t=1}^{T} log c_t \tag{3.18}$$

3.2.2.4. The Training Problem. The second problem is the training (learning) problem which we need to solve for recognition of facial expressions. Maximum likelihood is used where we would like to get Θ^* that maximizes the likelihood of the training sequences. The approach explained here is an Expectation Maximization (EM) procedure called the *Baum-Welch algorithm.* We start by defining two new variables: $\xi_t(i,j)$ as the probability of being in state i at time t and in state j at time $t+1$, given the observation sequence O and the model Θ; and $\gamma_t(i)$ as the probability of being in state i at time t, given the model Θ.

$$\xi_t(i,j) \equiv P(q_t = S_i, q_{t+1} = S_j | O, \Theta) \tag{3.19}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(l)} \tag{3.20}$$

$$\gamma_t(i) \equiv P(q_t = S_i|O, \Theta) \tag{3.21}$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} = \frac{\alpha_t(i)\beta_t(i)}{P(O|\Theta)} = \sum_{j=1}^{N} \xi_t(i,j) \tag{3.22}$$

EM procedure iteratively calls two steps, called the E-step and the M-step. In the E-step, $\gamma_t(i)$ and $\xi_t(i,j)$ are calculated from the current model parameters $\Theta = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ using Eq. 3.20 and Eq. 3.22. In the M-step, Θ is recalculated from these variables until the likelihood converges. If there are V sequences (videos) each with F_v observations (frames) in our training set, the elements of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Pi}$ are re-estimated in the M-step using the following equations:

$$\hat{a}_{ij} = \frac{\sum_{v=1}^{V} \sum_{t=1}^{F_v-1} \xi_t^v(i,j)}{\sum_{v=1}^{V} \sum_{t=1}^{F_v-1} \gamma_t^v(i)} \tag{3.23}$$

$$\hat{b}_j(m) = \frac{\sum_{v=1}^{V} \sum_{t=1}^{F_v-1} \gamma_t^v(j) 1(O_t^v = o_m)}{\sum_{v=1}^{V} \sum_{t=1}^{F_v-1} \gamma_t^v(j)} \tag{3.24}$$

$$\hat{\pi}_i = \frac{\sum_{v=1}^{V} \gamma_t^v(i)}{V} \tag{3.25}$$

3.2.2.5. Continuous Observation Densities in HMMs. In the discussion of HMMs, we assumed discrete observations modeled as a multinomial. But, if the inputs are continuous as the normalized landmark sequences in our case, one possibility is to discretize them (e.q. to motion vectors) and use these discrete observations. A vector quantizer that quantizes the motion of a landmark between each frame to one of the eight directions (right, up-right, up, up-left, etc.) can be a solution to discretize the landmark behaviours. A better way of quantization is to cluster the motion directions into K discrete observations using unsupervised clustering methods, such as K-means clustering.

Instead of quantizing the continuous observations into discrete ones, the trajectories of the landmarks can be assumed as normally distributed (or linear combinations of normal distributions) and we can model them using mixture of multivariate Gaussians.

Let there be K Gaussians used, which can be thought as the soft versions of K-means. Then, the probability of the component k in state i at time t given the model can be computed by:

$$\gamma_t(i,k) = \frac{\alpha_t(i)\beta_t(i)w_{ik}N(O_t;\mu_{ik},\Sigma_{ik})}{b_i(O_t)P(O|\Theta)} \tag{3.26}$$

where w_{ik} values are the weights of Gaussians. The calculated probabilities above will sum up to the the probability of being in state i at time t, given Θ as:

$$\gamma_t(i) = \sum_{k=1}^{K} \gamma_t(i,k) \tag{3.27}$$

For a sample, the M-step equations in this case will be:

$$\hat{\mu}_{ik} = \frac{\sum_{v=1}^{V}\sum_{t=1}^{F_v}\gamma_t(i,k)O_t}{\sum_{v=1}^{V}\sum_{t=1}^{F_v}\gamma_t(i,k)} \tag{3.28}$$

$$\hat{\Sigma}_{ik} = \frac{\sum_{v=1}^{V}\sum_{t=1}^{F_v}\gamma_t(i,k)(O_tO_t^T - \hat{\mu}_{ik}\hat{\mu}_{ik}^T)}{\sum_{v=1}^{V}\sum_{t=1}^{F_v}\gamma_t(i,k)} \tag{3.29}$$

3.2.2.6. HMMs in Expression Recognition. As explained in the normalization of tracked facial landmarks, the samples in the training set are frame sequences where each frame is a vector consisting of normalized landmark locations. So, we use Continuous Hidden Markov Models (CHMM) with multivariate samples.

Let there be C different expression classes that we want to recognize in our application. First, a model Θ_c is trained from the training set of each class. Then, when a new observation O comes, the class which is giving the maximum likelihood is assigned as its class, that is c_O.

$$c_O = \arg\max_c P(O|\Theta_c) \tag{3.30}$$

4. EXPERIMENTS AND RESULTS

4.1. Database

Since the aim of this thesis is to classify the most common non-manual signs (head motion and facial expressions) in Turkish Sign Language (TSL), we collected a video database of non-manual signs to experiment the proposed approach and compare it with the other types of Multi-resolution Active Shape Model (MRASM) trackers.

The non-manual signs which are frequently used in TSL and those changing the meaning of the performed sign considerably are selected as the sign classes in the database. There are also additional signs which we use in daily life during speaking. This database was collected and first presented in 2007 [43]. The database involves 11 (6 female, 5 male) different subjects performing 8 different classes of signs each.

4.1.1. The non-manual signs used in the database

Some of the selected signs involve only head motion or facial expressions and some involve both. So, we use *sign* and *expression* terms to refer to a class we use in the database. The database is formed of the following 8 different classes of signs:

1. *Neutral*: The neutral state of the face. The subject neither moves his/her face nor makes any facial expressions.
2. *Head L-R*: Shaking the head to right and left sides. The initial side varies among subjects, and the shaking continues about 3-5 times. This sign is frequently used for negation in TSL.
3. *Head Up*: Raise the head upwards while simultaneously raising the eyebrows. This sign is also frequently used for negation in TSL.
4. *Head F*: Head is moved forward accompanied with raised eyebrows. This sign is used to change the sentence into a question form in TSL. It resembles the surprise expression used in daily life.

5. *Sadness*: Lips turned down, eyebrows down. It is used to show sadness, e.g. when apologizing. Some subjects also move their head downwards.
6. *Head U-D*: Nodding head up and down continuously. Frequently used for agreement.
7. *Happiness*: Lips turned up. Subject smiles.
8. *Happy U-D*: Head U-D + Happiness. The preceding two classes are performed together. It is introduced to be a challenge for the classifier in successfully distinguishing this confusing class with the two preceding ones.

In Figure 4.1, some frames captured from different sign classes can be seen.

4.1.2. Properties of the Database

- It involves 11 different subjects (6 female, 5 male).
- Each subject performs 5 repetitions for each of 8 classes. So there are a total number of 440 videos in the database.
- Each video lasts about 1-2 seconds.
- Philips SPC900NC web cam is used with choice of 640×480 resolution and 30 fps.
- The recording is done in a room eliminated from sunlight and illuminated by using daylight halogen and fluorescent lights.
- The videos are compressed with “Indeo 5.10” video codec.
- Each video starts in neutral state, the sign is performed and again ends in neutral state.
- No subjects have beard, moustache or eyeglasses.
- There is no occlusion or motion blur.

4.1.3. Annotated Videos

In order to satisfy different experiments on the database, a preferably large number of facial landmarks are chosen for manual annotation. The selected 60 points can be seen in Figure 4.2. Due to the difficulty in manually annotating these landmarks

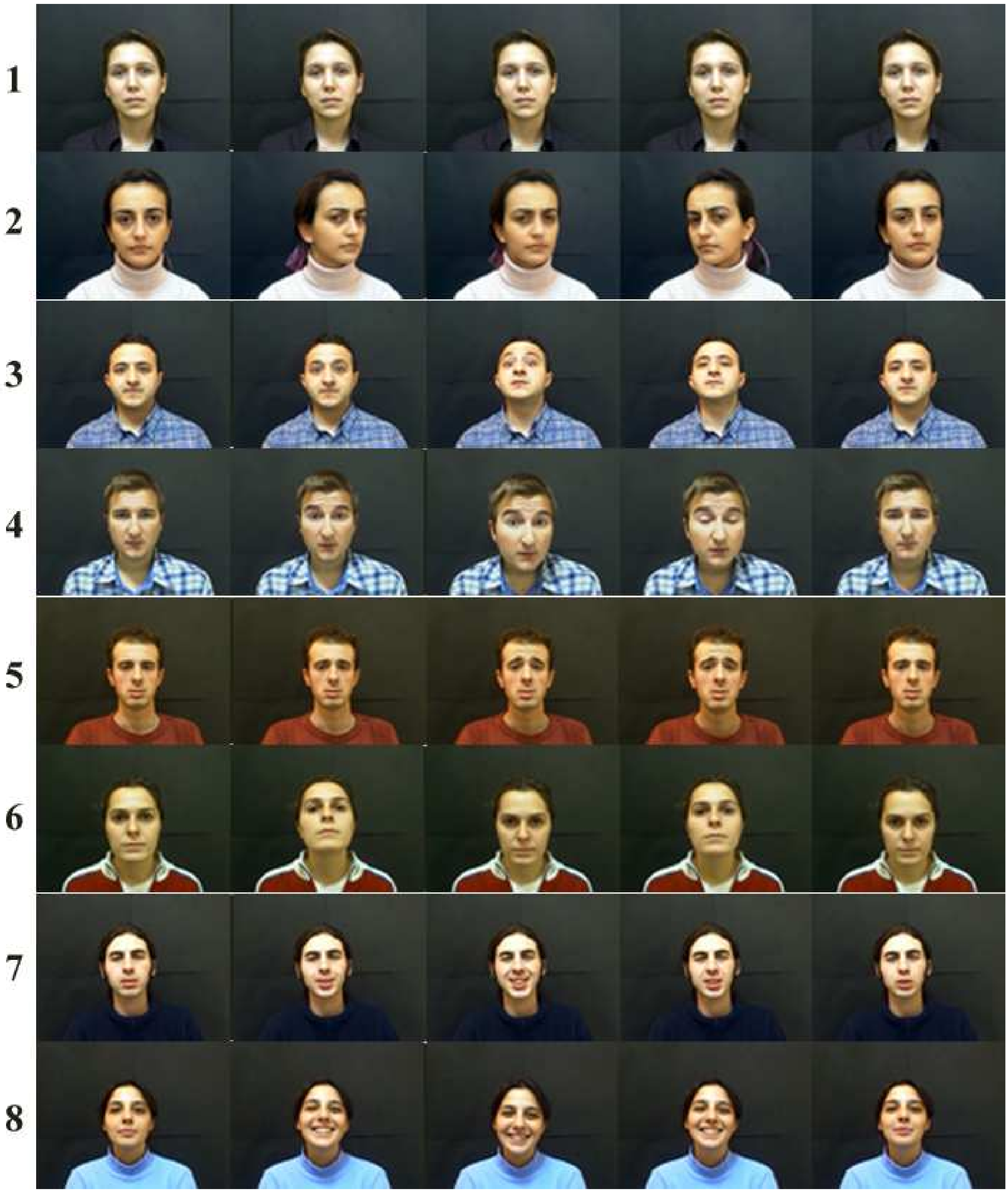


Figure 4.1. 8 different non-manual sign classes are used in the database

in all frames, only 3 repetitions of 4 classes (Head L-R, Head Up, Head F, Happiness) performed by 4 subjects (2 male, 2 female) are annotated in the database. So, there are a total of 48 annotated videos. In total 2880 (48 videos × 60 average frames per video) frames are annotated.

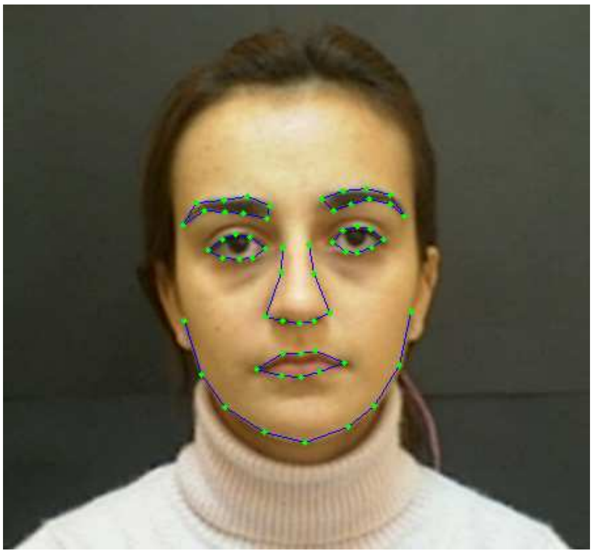


Figure 4.2. 60 facial landmarks are selected for ground truth in the videos.

4.1.4. Annotated Images

Experimentally, it was observed that more landmarks would lead to better ASM tracking. Thus, we increased the landmarks and 116 feature points (eyes: 24, eyebrows: 28, eye lids: 14, wrinkles below eyes: 14, nose peak: 5, lips: 16, chin: 15) were selected for annotation in each frame as seen in Figure 4.3.

For each of the 4 subjects that we selected for video annotation in the previous subsection, random frames were captured from all videos (8 different classes of signs) and approximately 60 of these frames (30 frontal, 10 upwards, 10 left and 10 right) were selected for annotation. Sample frames are seen in Figure 4.4.

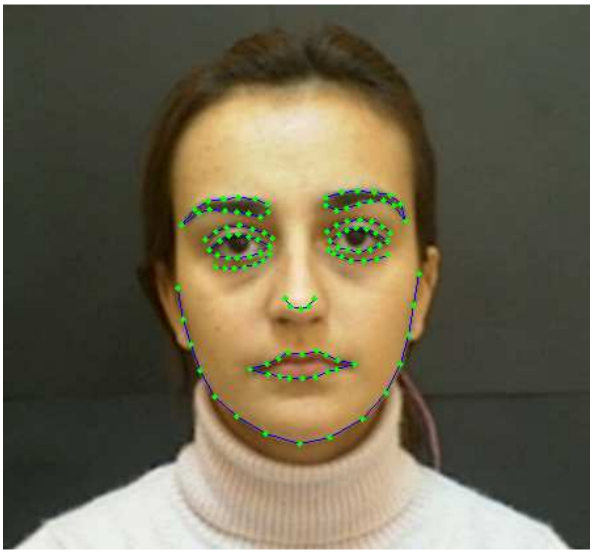


Figure 4.3. 116 facial landmarks are selected for ground truth in the captured images.

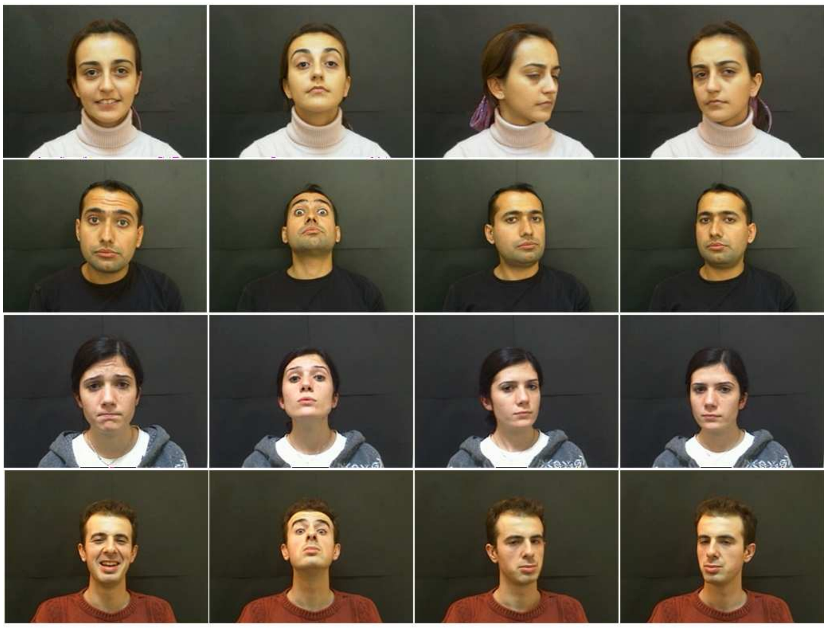


Figure 4.4. About 60 frames were selected from 4 different subjects and each set is divided into one of the 4 different views (frontal, upwards, left or right).

4.2. Tracking Experiments and Results

4.2.1. Experiment Setup

Since there exists human bias in the manual annotation of the landmarks, data refitting is performed on the training set. That is; a one level ASM model is trained from the training set. Then, for each sample in the training set, the annotated shape is set as the initial shape and one level person specific ASM fitting is performed on this sample. The new locations of the landmarks are accepted as the ground truth and the model used for further testing is trained from these locations. For example, for a subject with 35 frontal training images, the mean Euclidean distance between the ground truth and the best fit landmark locations for each image before data refitting and after data refitting are shown in Figure 4.5. The mean Euclidean distance of all images decreased from 2.13 to 1.73 pixels when data refitting is applied.

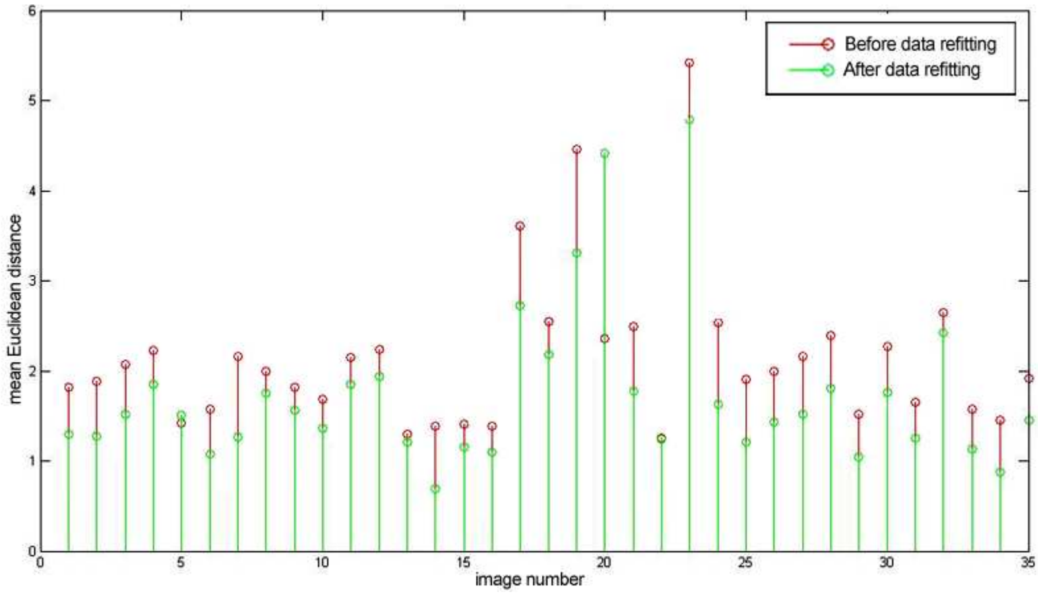


Figure 4.5. Data refitting comparison

Afterwards, in order to compare the multi-view vs. single view and person specific vs. generic MRASM (Multi-resolution ASM) tracking, we abbreviated 4 different types of MRASM training as in Table 4.1:

Table 4.1. Types of MRASM training

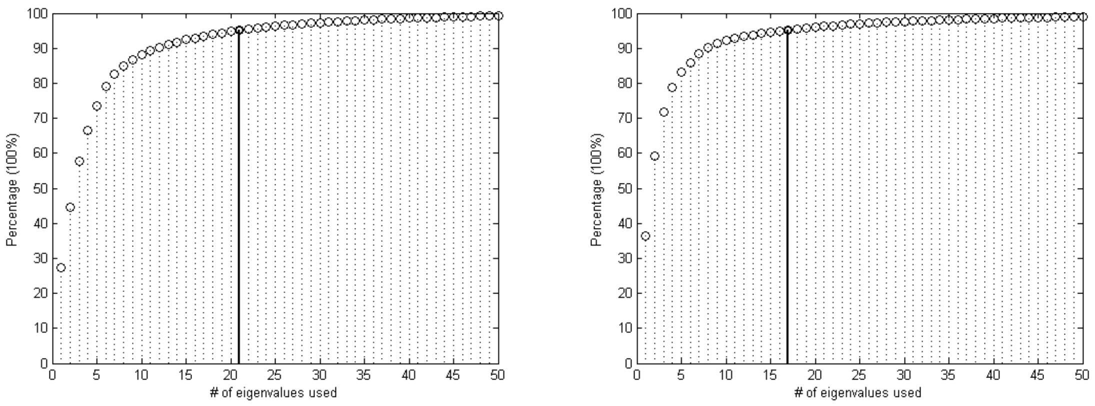
	Person specific	**Generic**
Multi-view	M-view P-s (Type I)	M-view G (Type II)
Single view	S-view P-s (Type III)	S-view G (Type IV)

In multi-view training, a different model for each of the views (frontal, upwards, left and right) was trained whereas a single model is trained from all views combined together in the single-view approach. In the person-specific approach, the subject is known and tracked using its own model. But in the generic approach, a unified model is trained from all subjects and this model is used for tracking landmarks of any subject.

4.2.2. Statistical Analysis Results

4.2.2.1. Sufficient number of eigenvectors to describe most of shape variance. We first determine the number of eigenvectors to be retained: Figure 4.6 shows the variance retained versus the number of eigenvectors. A small number of (15-20) eigenvectors (about 8-10% of all eigenvectors) seemed to be enough to describe 95% of the total shape variation in the training data set. One expects to have fewer eigenvectors when only one view (e.g. frontal) is used instead of all views, but the plots in Figure 4.6 show that when a percent of total variation is considered, the sufficient number of eigenvectors depends on the distribution of the variation along the eigenvectors of the used sample set. In other words, when we use 95% of the total variation in all views where the variation in head movement dominates the sample set, we may lose facial expression changes such as smiling. But the multi-view approach (dividing the sample set into similar subsets) enables us to model these changes better.

4.2.2.2. How does the shape change in the most significant eigenvectors?. Figure 4.7 shows the variation of the shape along the most significant three eigenvectors. The most variation was along the left to right head shaking when we analyzed all training shapes. Other significant movements were head nodding and orientation with a surprise



(a) (b)

Figure 4.6. Eigenvector contributions to the total variance in shape space (a) if a generic frontal model is used and (b) if a generic single (unified) view model is used.

expression as seen in Figure 4.7.

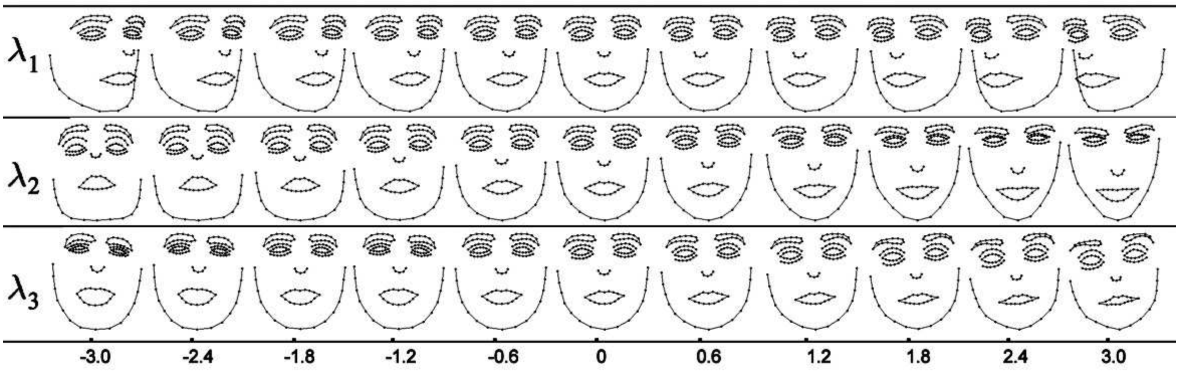


Figure 4.7. The variation along the most significant 3 eigenvectors in the shape space

4.2.2.3. Sufficient number of eigenvectors to describe most of the combined variance. When shape and texture data are analyzed together, 82 eigenvectors were sufficient to describe 95% of the total variation as seen in Figure 4.8.

4.2.2.4. The effect of photometric normalization. In Figure 4.9, the effect of reducing global illumination by photometric normalization is shown. It is clearly seen that the illumination variation in the normalized images below is less than the corresponding

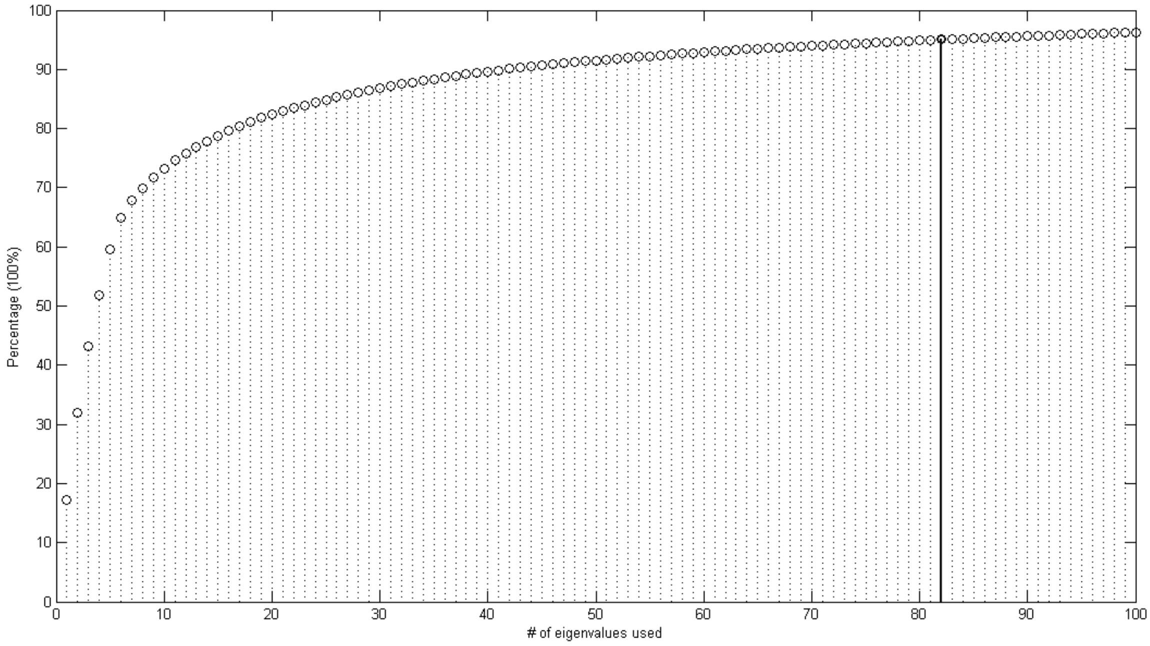


Figure 4.8. Eigenvector contributions to the total variance in appearance space

original images above. Especially, the skin color of the subject with the 3^{rd} and 4^{th} faces from left seems to change considerably.

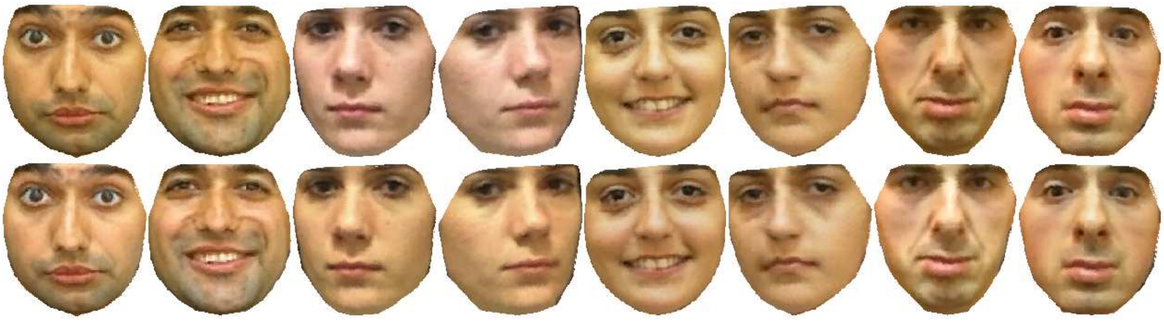


Figure 4.9. The effect of photometric normalization (original textures above and corresponding normalized ones below)

4.2.2.5. How does the shape/appearance change in the most significant eigenvectors?. The variation in the significant eigenvectors in this combined model can be seen in Figure 4.10. The analysis results show that the person specific texture properties, left to right head shaking and happiness vs. sadness dominated the total variance in the data set. Also notice that, there is no significant illumination variance observed in Figure 4.10 as a result of photometric normalization.

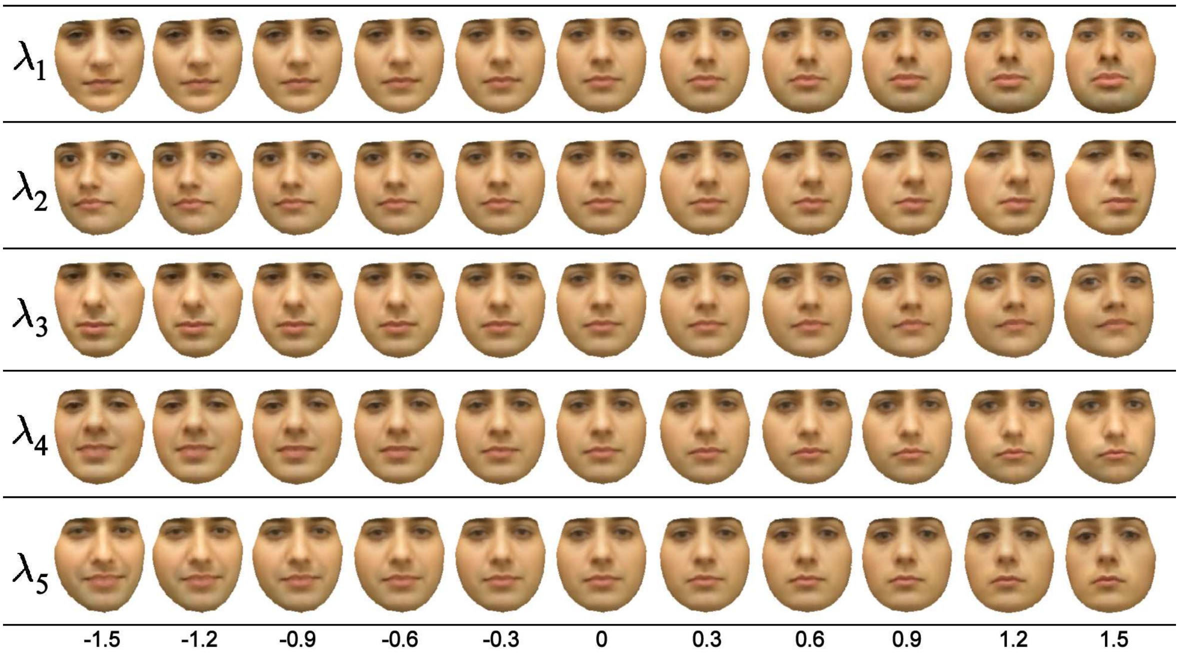


Figure 4.10. The variation along the most significant 5 eigenvectors in the combined shape and texture space

<u>4.2.2.6. The effect of training set selection on the principle variation modes.</u> In [43], the distinction between the eigenvectors was clearer and more interpretable since the training images included 11 subjects instead of 4 subjects. The more the number of training subjects are used, the better the distinction of appearance between eigenvectors is achieved and the better analysis is done in terms of observing head movement and facial expression changes. This fact shows the importance of training set selection when one wants to interpret the appearance values.

4.2.3. Comparison of ASM and Multi-resolution ASM

In Figure 4.11, the face shapes which are found by regular (only 1-level) ASM search after 30 iterations and by 3-level MRASM search after a maximum of 5 iterations allowed in each level are shown. Convergence ratio is taken as 0.9. m and n are chosen as (17,13,13) and (13,9,9) for each level l ($l = 0, 1, 2$) respectively. It is seen in the figure that multi-resolution increases the performance. MRASM search can fit well where ASM converges to local minima. Furthermore, MRASM search with the choices described above was two times faster than the regular ASM search.

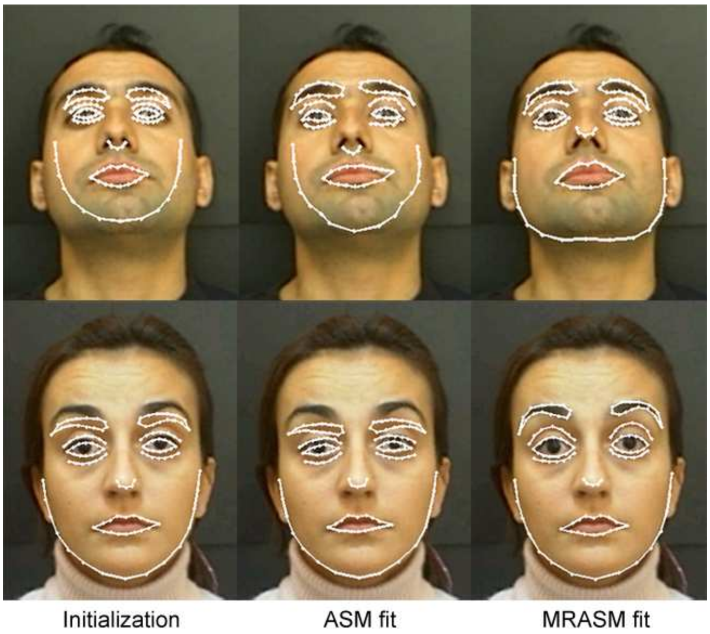


Figure 4.11. Comparison of ASM fit and MRASM fit

An example search using MRASM is shown in Figure 4.12 where 5 iterations are performed at each level. The sizes of the images in level 2 and level 1 are quarter and half of the the original images in level 0 respectively. But they are scaled to the same size in the figure for illustration. It is seen that the shape is roughly fitted in the top level and fine tuned in the following levels.

4.2.4. Tracking Results

Sample tracking results with person specific multi-view tracker can be seen in Figure 4.13 where the index on the top right stands for the frame number in that video.

Due to the difficulty in reporting all of the tracking results of different types of

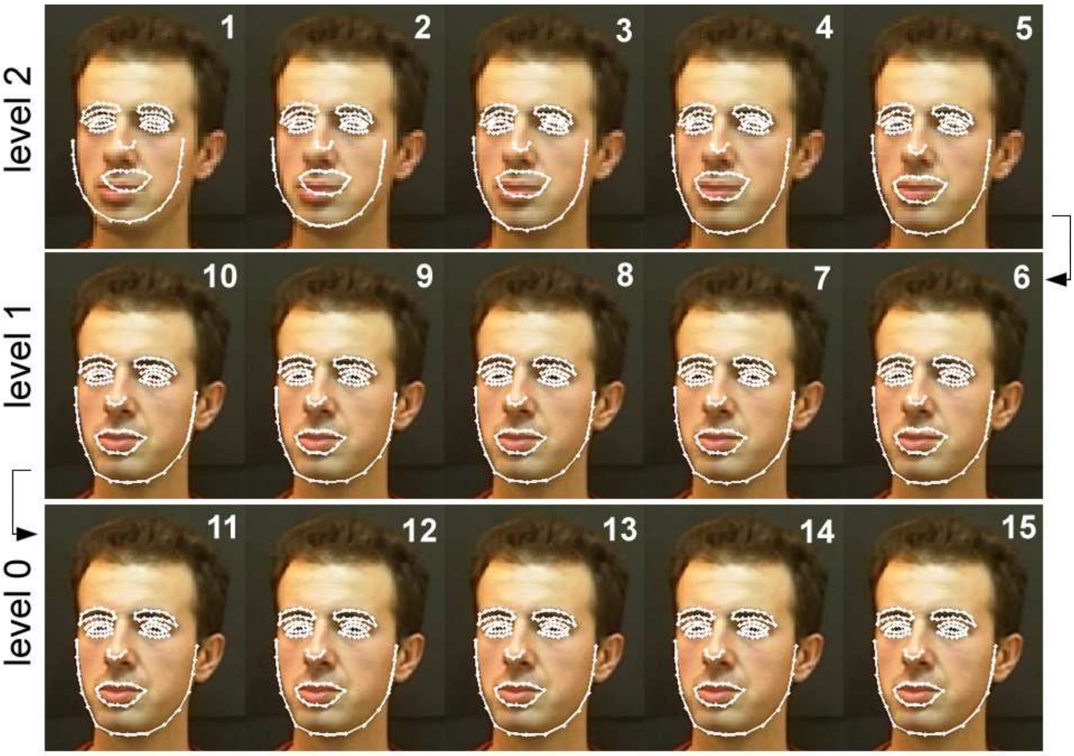
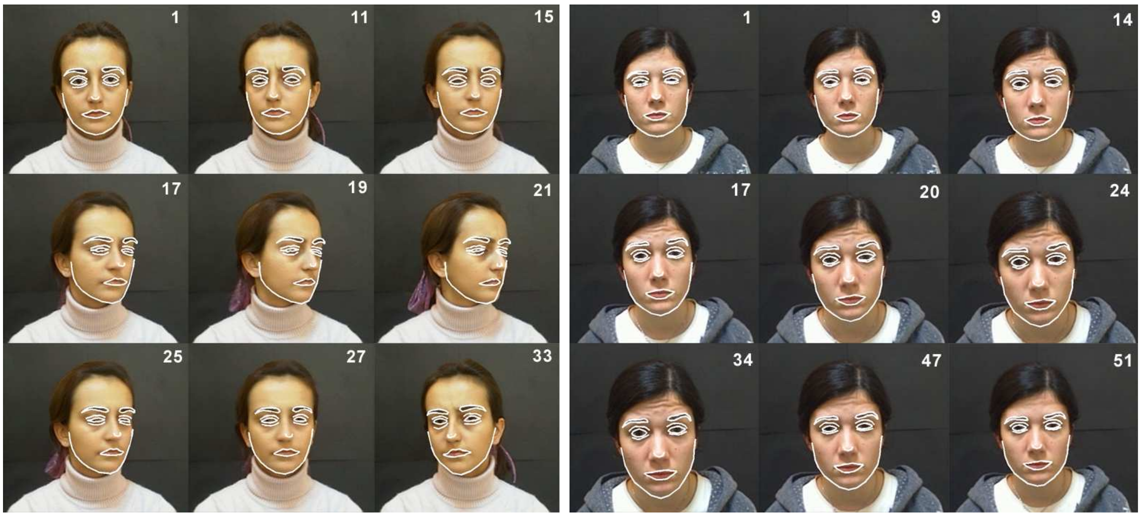


Figure 4.12. Searching using Multi-resolution ASM



(a)

(b)

Figure 4.13. Found landmarks in a video that belongs to (a) 2^{nd} class and (b) 3^{rd} class

experiments given in Table 4.1 visually, we take the landmarks in annotated video frames as the ground-truth data and report the Euclidean distance between annotated landmarks and the tracked landmarks as the error metric. But, there is inconsistency between the 60 ground truth landmarks per frame (Section 4.1.3) and the 116 tracked landmarks which are found by MRASM tracker that is trained from annotated images (Section 4.1.4). So, we formed a mapping from ground truth and tracked data to the new 52 feature points which are seen in Figure 4.14 to make the landmarks comparable. In addition, each sequence of annotated and tracked shapes is re-sampled to 60 frames so as to equalize the frame length and be able to take the average of the error in that frame over many samples. A total of 48 videos (4 subjects, 4 classes, 3 repetitions per class) are analyzed and different types of resulting errors are reported in the following subsections. All the Euclidean distances (errors) are in pixels where each frame has a resolution of 640 × 480 pixels.

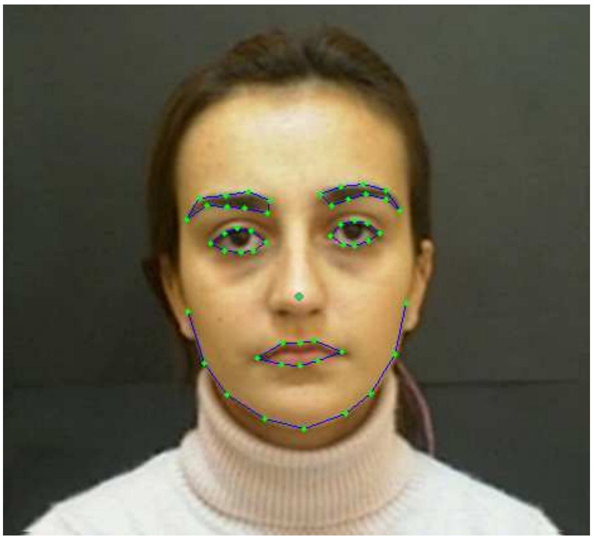
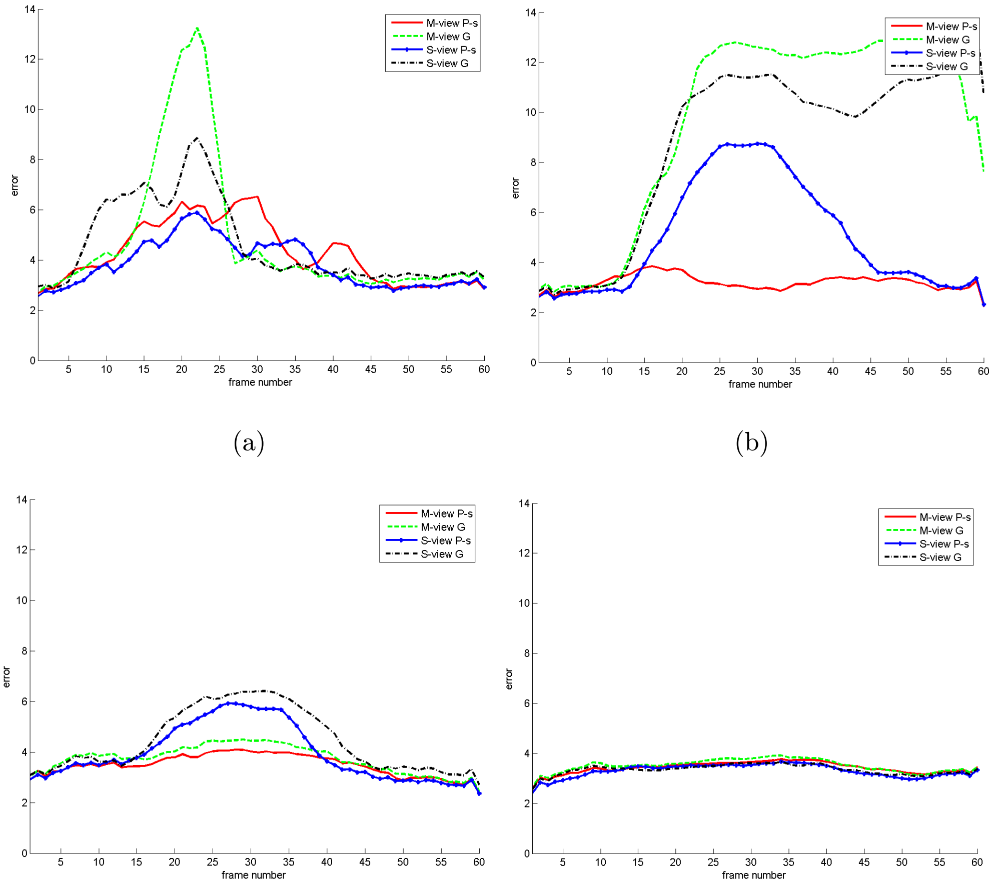


Figure 4.14. Common 52 facial landmarks are selected in order to compare ground truth and tracked landmarks.

4.2.4.1. Tracking Performance of Each Class. In Figure 4.15, the mean error found by four different tracking approaches (Table 4.1) is plotted for each class. When all classes are considered, it is observed that the multi-view person specific tracker performs the best. Especially, it is the only one that can track the landmarks in *Head Up* sign.



(a)

(b)

(c)

(d)

Figure 4.15. Mean Euclidean distance between ground truth and tracked landmarks for each class which are (a) Head L-R, (b) Head Up, (c) Head F, and (d) Happiness.

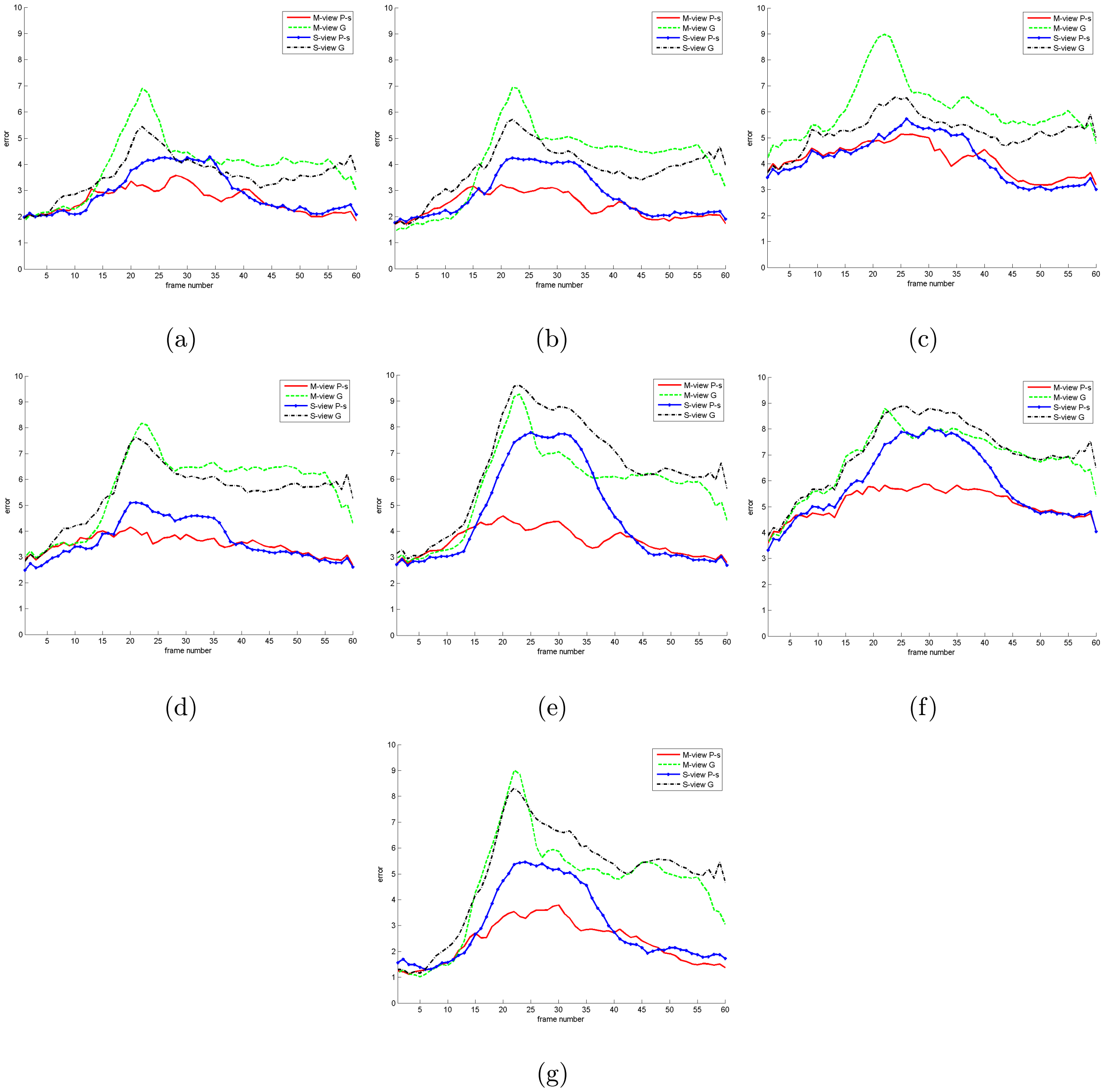
4.2.4.2. Tracking Performance on Each Face Component. Similarly, mean error for each component of the face is plotted in Figure 4.16. The reported components are left/right eye, left/right eyebrow, lips, chin and nose tip of the face of the subject as shown in Figure 4.14. It is seen that multi-view person specific tracking outperforms all other trackers in all components. We observed that lips are the most difficult component to track because they are the most deformable part of the face and the intensity contrast between the skin color and lips color is not very high and varies among subjects.

4.2.4.3. Comparison of Trackers . It can be said that person specific tracking performs better than the generic approaches. The multi-view approach in person specific tracking increases the performance whereas it results in a decrease in the generic approach.

4.2.4.4. Mean Error in a Video with Corresponding Landmarks. As mentioned earlier, the mapping from 116 tracked landmarks and from 60 ground truth landmarks to 52 new landmarks is done approximately. When the error plots are considered with the corresponding tracking results visually, it can be said that 2-3 pixels of mean Euclidean distance comes from this approximation. It mainly stems from the fact that the annotators were different for ground truth videos and training images.

Furthermore, a user interface that eases the video annotation by copying and pasting landmarks between consecutive frames and then dragging them is implemented and used. As a drawback of this convenient facility, the locations of the landmarks remain constant where they should change in some frames. So, the ground truth data includes not exactly but roughly correct locations of landmarks.

In order to show this, the annotated and finely tracked shapes are given for selected frames of the same video file with the corresponding error plot in Figure 4.17. It can be observed that the tracked shape in Figure 4.17b is clearly more accurate than the annotated shape in Figure 4.17a. The error in Figure 4.17c reflects this discrepancy.



(a)

(b)

(c)

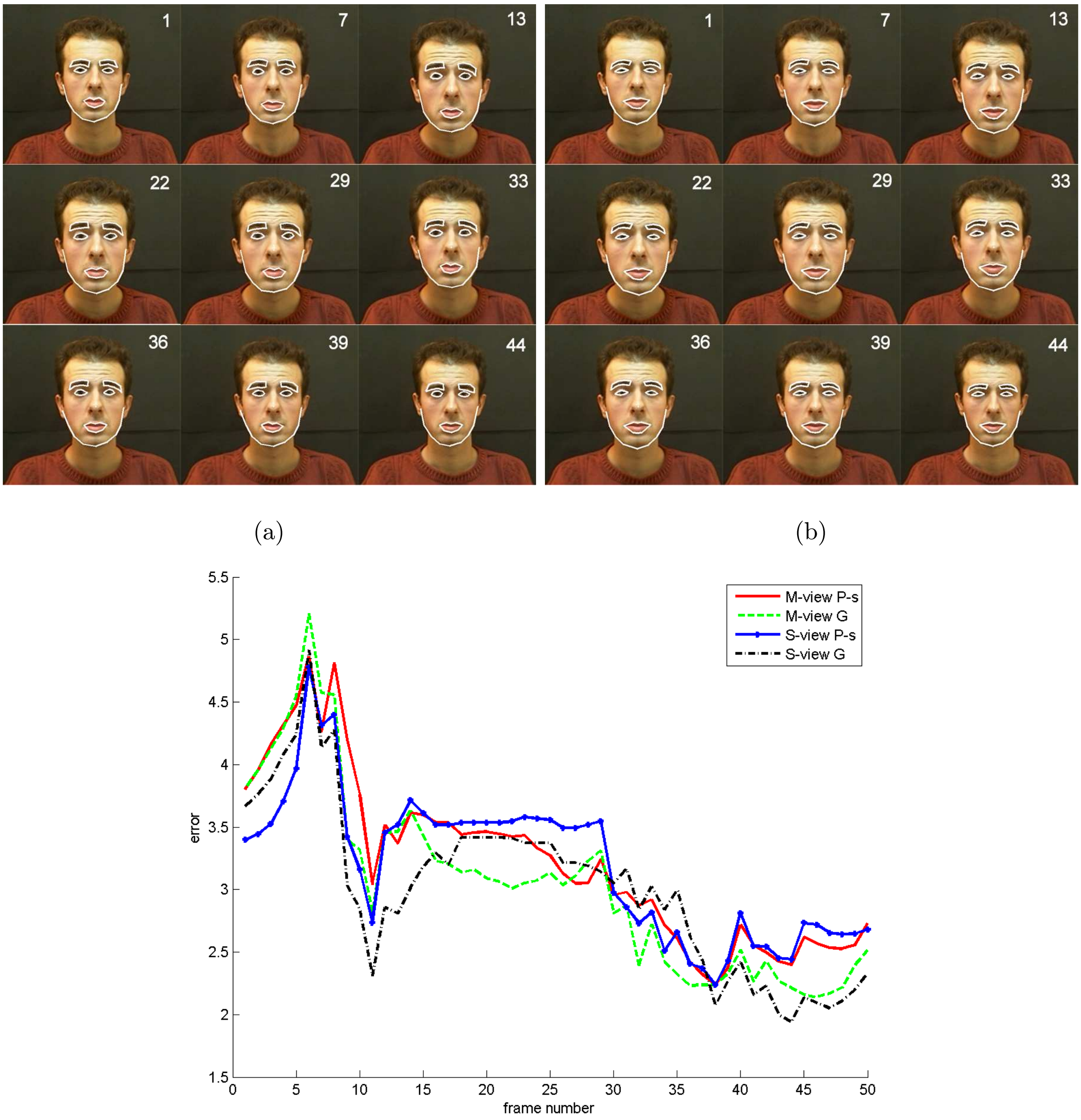
(d)

(e)

(f)

(g)

Figure 4.16. Mean Euclidean distance between ground truth and tracked landmarks in each frame for each component. Components are (a) left eye, (b) right eye, (c) left eyebrow, (d) right eyebrow, (e) lips, (f) chin and (g) nose tip of the subject.



(a)

(b)

(c)

Figure 4.17. (a) The ground truth landmarks, (b) the tracked landmarks with M-view P-s tracker and (c) corresponding error plot.

As described in Chapter 2, the frames are left empty and then interpolated if the best fitted shape is not accepted as valid. By this method, the tracker can be re-initialized and thus, the correct landmark locations can be re-caught after a few frames. In Figure 4.18 and Figure 4.19, it can be observed that the single view person specific tracker stays unchanged for a few frames, then tracks the landmarks again. Furthermore, the difference between ground truth landmarks and correctly tracked landmarks can be seen from the sub-figures (a) and (b) in both figures.

4.3. Expression Recognition Experiments and Results

4.3.1. Experiment Setup

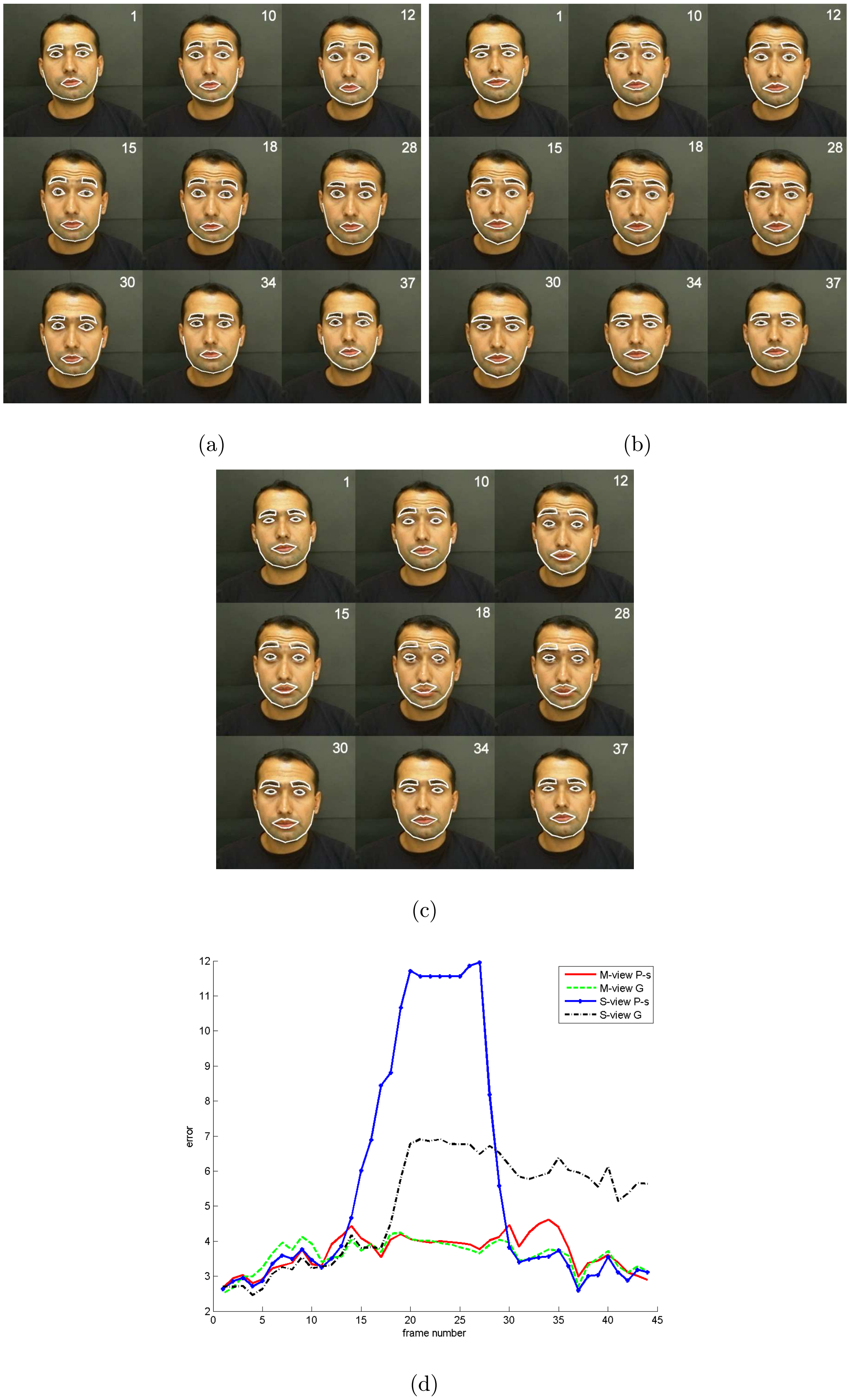
We performed the classification experiments using the multi-variate continuous HMM classifier [60]. The number of Gaussians and the number of hidden states are taken as 2 and 6, respectively. We prepared the following sets for our experiments where each shape sequence (video) is normalized:

- $\mathbf{\Phi_{1:4}}$: Involves 7 classes and 5 repetitions for 4 subjects found with the tracker. All classes except *Neutral* are included. (140 samples)
- $\mathbf{\Phi_{gt}}$: Involves 4 subjects and 3 repetitions for each of 4 classes in the ground truth data. The classes are *Head L-R*, *Head Up*, *Head F* and *Happiness.* (48 samples)

Then we designed four tests on these sets:

<u>4.3.1.1. Test I.</u> Use 2 repetitions for each class of each subject in $\mathbf{\Phi_{gt}}$ for training and the remaining one repetition for testing. Cross validation is performed by leaving one repetition out . (32 training samples, 16 test samples)

<u>4.3.1.2. Test II.</u> Use 3 repetitions of each class of each subject in $\mathbf{\Phi_{1:4}}$ for training and the remaining 2 repetitions for testing by leave-two-repetitions-out cross validation. (84 training samples, 56 test samples)



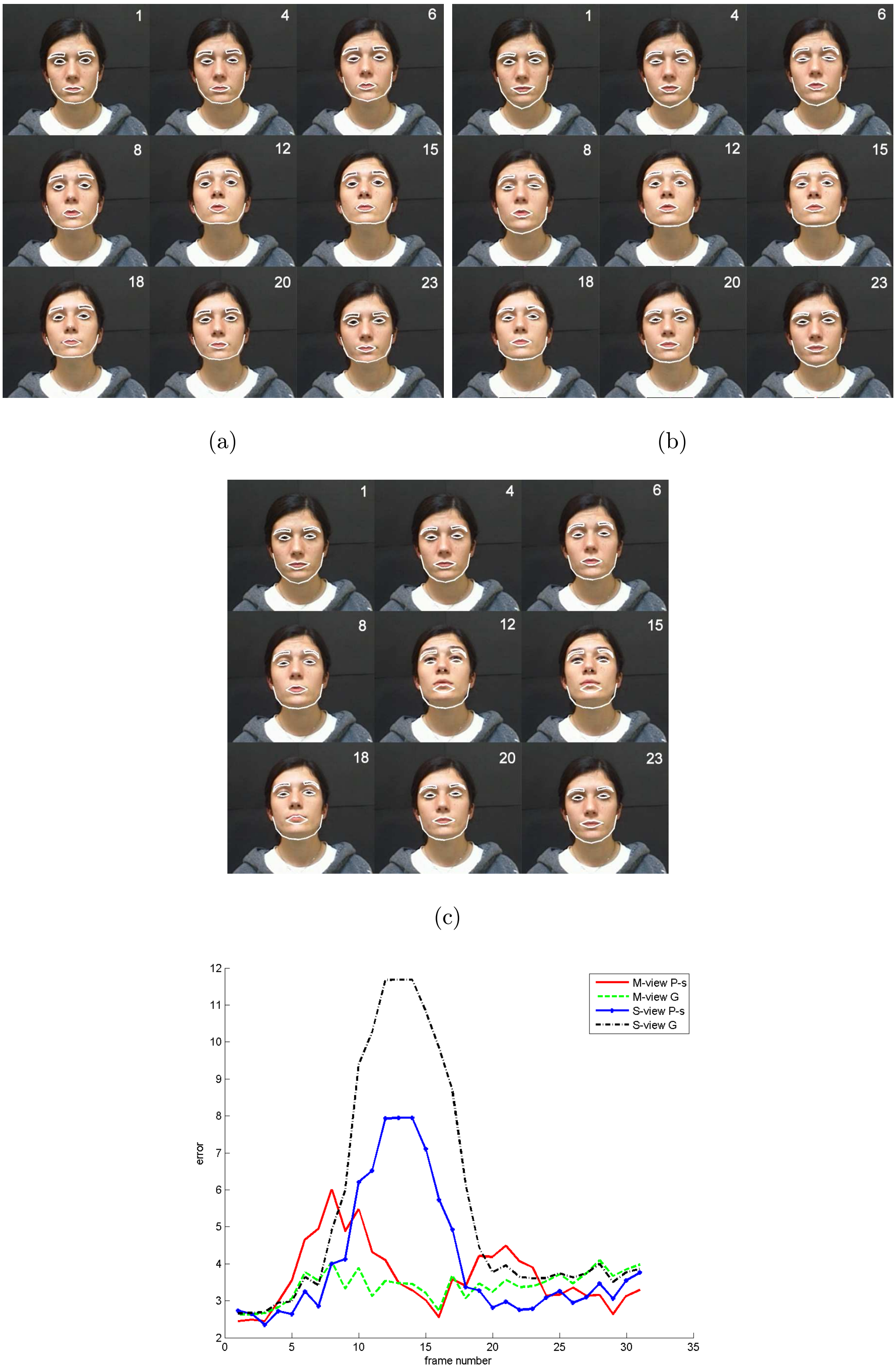
(a)

(b)

(c)

(d)

Figure 4.18. (a) The ground truth landmarks, (b) the tracked landmarks with multi-view generic tracker, (c) the tracked landmarks with single view person specific tracker, and (d) corresponding error plot for a video in *Head F* class.



(a)

(b)

(c)

(d)

Figure 4.19. (a) The ground truth landmarks, (b) the tracked landmarks with multi-view generic tracker, (c) the tracked landmarks with single view generic tracker, and (d) corresponding error plot for a video in *Head Up* class.

4.3.1.3. Test III. Use all samples of 3 subjects in $\mathbf{\Phi_{1:4}}$ for training and test on the unseen subject. It is performed 4 times by leaving each subject out for testing, that is leave-one-subject-out cross validation. (105 training samples, 35 test samples)

4.3.1.4. Test IV. 3 repetitions are selected from $\mathbf{\Phi_{gt}}$ for training and testing is done on the remaining 2 repetitions in $\mathbf{\Phi_{1:4}}$. (48 training samples, 32 test samples)

The tests are done using all the trackers given in Table 4.1 and the results are described in the following subsection.

4.3.2. Recognition Results

The accuracy results found for each test and with each tracker type are given in Table 4.2.

It is observed that continuous HMM classifier accurately classifies the ground truth videos into the correct classes when 4 classes are considered (Test I). Additionally, training with ground truth data and testing on the unseen videos can be performed with 100% success rate when we use the multi-view person specific tracker (Test IV). If the number of classes is increased to 7, the best accuracy when we use some of the repetitions for training and the remaining ones for testing is about 85% (Test II). When the testing is done on an unseen subject where the tracker is trained from the other subjects, we can achieve 73% accuracy (Test III). In all cases, multi-view person specific tracker results seem to be the most reliable ones. But, in the absence of the subject information (i.e. when we need a generic tracker), multi-view MRASM tracking results seem to be near to the best results achieved.

In addition, the confusion matrices (in percentage) for each test are given in Tables 4.3, 4.4 and 4.5. The confusion matrix for the first test is not given because the accuracy is 100%.

Table 4.2. Expression classification results

Test no	# classes	# training samples	# test samples	Tracker type	Accuracy (%)
I	4	32	16	Ground truth	**100.00**
II	7	84	56	M-view P-s	**84.82**
				M-view G	83.57
				S-view P-s	81.79
				S-view G	73.93
III	7	105	35	M-view P-s	**72.86**
				M-view G	68.57
				S-view P-s	64.29
				S-view G	53.57
IV	4	48	32	M-view P-s	**100.00**
				M-view G	93.75
				S-view P-s	93.75
				S-view G	87.50

An important result which is observed in confusion matrices is that *Happy U-D* is misclassified as *Head U-D* in some of the videos. This shows that the head motion dominates the classification decision and the classifier may be incapable of catching the change in facial expression in the existence of large head movements. In addition, only the perimeter of the lips includes landmarks and there are no landmarks inside the lips. This decreases the interpretability of the emotional expression when the head also moves because the area, perimeter length or other properties of lips vary among subjects and it may not be clear that the mouth is open or not. If *Happy U-D* and *Head U-D* classes were considered as the same class in the outputs, the best accuracy would be 92% and 81%, for tests II and III, respectively.

Moreover, *Sadness* seems to be a difficult class to correctly classify. This mainly results from the fact that the performance of sadness varies among subjects. Some of them also move their head in addition to turning down the lips. This movement confuses the classifier.

Table 4.3. Confusion matrices for test II found with (a) multi-view person specific, (b) multi-view generic, (c) single view person specific and (d) single view generic trackers.

(a)

	Head L-R	Head Up	Head F	Sadness	Head U-D	Happiness	Happy U-D
Head L-R	100.0	0	0	0	0	0	0
Head Up	0	100.0	0	0	0	0	0
Head F	0	0	98.75	0	0	0	1.25
Sadness	7.5	0	0	56.25	21.25	5.0	10.0
Head U-D	0	0	0	0	95.0	0	5.0
Happiness	6.25	0	0	1.25	0	83.75	8.75
Happy U-D	1.25	0	0	0	38.75	0	60.0

(b)

	Head L-R	Head Up	Head F	Sadness	Head U-D	Happiness	Happy U-D
Head L-R	98.75	0	0	0	0	0	1.25
Head Up	0	97.5	0	0	2.5	0	0
Head F	0	6.25	91.75	0	0	2.5	0
Sadness	3.75	0	5.0	52.5	22.5	5.0	11.25
Head U-D	0	3.75	0	0	93.75	0	2.5
Happiness	0	0	5.0	0	0	90.0	5.0
Happy U-D	1.25	5.0	0	0	32.5	0	61.25

(c)

	Head L-R	Head Up	Head F	Sadness	Head U-D	Happiness	Happy U-D
Head L-R	100.0	0	0	0	0	0	0
Head Up	1.25	90.0	0	0	0	8.75	0
Head F	0	0	95.0	5	0	0	0
Sadness	11.25	0	2.5	48.75	22.5	5	10
Head U-D	0	0	0	0	93.75	0	6.25
Happiness	11.25	0	0	1.25	0	83.75	3.75
Happy U-D	1.25	0	0	0	37.5	0	61.25

(d)

	Head L-R	Head Up	Head F	Sadness	Head U-D	Happiness	Happy U-D
Head L-R	95.0	0	0	0	2.5	0	2.5
Head Up	0	95.0	0	0	0	5.0	0
Head F	0	7.5	87.5	5.0	0	0	0
Sadness	16.25	1.25	0	32.5	17.5	8.75	23.75
Head U-D	0	18.75	0	0	71.25	0	10.0
Happiness	7.5	0	0	1.25	0	81.25	10.0
Happy U-D	1.25	12.5	3.75	0	27.5	0	55.0

Table 4.4. Confusion matrices for test III found with (a) multi-view person specific, (b) multi-view generic, (c) single view person specific and (d) single view generic trackers.

(a)

	Head L-R	Head Up	Head F	Sadness	Head U-D	Happiness	Happy U-D
Head L-R	100.0	0	0	0	0	0	0
Head Up	0	100.0	0	0	0	0	0
Head F	0	0	80.0	15.0	0	0	5.0
Sadness	10.0	0	0	55.0	15.0	5.0	15.0
Head U-D	0	0	0	0	95.0	0	5.0
Happiness	25.0	0	0	10.0	0	60.0	5.0
Happy U-D	25.0	0	0	5.0	50.0	0	20.0

(b)

	Head L-R	Head Up	Head F	Sadness	Head U-D	Happiness	Happy U-D
Head L-R	100.0	0	0	0	0	0	0
Head Up	0	75.0	0	0	25.0	0	0
Head F	0	20.0	75.0	5.0	0	0	0
Sadness	5.0	0	5.0	55.0	5.0	5.0	25.0
Head U-D	0	0	0	0	100.0	0	0
Happiness	15.0	15.0	5.0	0	0	60.0	5.0
Happy U-D	25.0	0	15.0	0	45.0	0	15.0

(c)

	Head L-R	Head Up	Head F	Sadness	Head U-D	Happiness	Happy U-D
Head L-R	100.0	0	0	0	0	0	0
Head Up	5.0	90.0	0	0	0	5.0	0
Head F	5.0	5.0	75.0	5.0	10.0	0	0
Sadness	50.0	0	0	15.0	10.0	5.0	20.0
Head U-D	0	0	0	0	90.0	0	10.0
Happiness	30.0	0	0	5.0	0	60.0	5.0
Happy U-D	25.0	0	0	5.0	50.0	0	20.0

(d)

	Head L-R	Head Up	Head F	Sadness	Head U-D	Happiness	Happy U-D
Head L-R	95.0	0	0	0	0	0	5.0
Head Up	0	75.0	0	0	10.0	5.0	10.0
Head F	0	15.0	60.0	5.0	10.0	5.0	5.0
Sadness	40.0	0	0	20.0	5.0	10.0	25.0
Head U-D	5.0	25.0	5.0	0	60.0	0	5.0
Happiness	25.0	0	5.0	15.0	0	55.0	0
Happy U-D	25.0	25.0	15.0	0	25.0	0	10.0

Table 4.5. Confusion matrices for test IV found with (a) multi-view person specific, (b) multi-view generic, (c) single view person specific and (d) single view generic trackers.

(a)

	Head L-R	Head Up	Head F	Happiness
Head L-R	100.0	0	0	0
Head Up	0	100.0	0	0
Head F	0	0	100.0	0
Happiness	0	0	0	100.0

(b)

	Head L-R	Head Up	Head F	Happiness
Head L-R	100.0	0	0	0
Head Up	0	75.0	25.0	0
Head F	0	0	100.0	0
Happiness	0	0	0	100.0

(c)

	Head L-R	Head Up	Head F	Happiness
Head L-R	100.0	0	0	0
Head Up	0	87.5	0	12.5
Head F	0	0	100.0	0
Happiness	12.5	0	0	87.5

(d)

	Head L-R	Head Up	Head F	Happiness
Head L-R	87.5	0	12.5	0
Head Up	0	75.0	12.5	12.5
Head F	0	0	100.0	0
Happiness	12.5	0	0	87.5

When compared to our previous joint-work with Asli Uyar [34], the proposed CHMM-based classifier performs better than the previously proposed SVM-based classifier. This result is achieved by comparing the ground truth classification accuracies of both studies because other tests are not convenient for comparison. The higher performance of CHMM approach is two-fold we suppose: First, the feature extraction by normalization of landmark locations increased the information about the performed sign instead of using maximum displacement vectors. Second, HMMs are widely accepted for their power in classifying sequential data as in image sequences of non-manual gestures, where the length of each sample is not to be normalized.

5. CONCLUSIONS

In this study, we investigated facial feature tracking in image sequences, and recognition of the performed non-manual gestures which are composed of head movements or facial expressions. Multi-resolution Active Shape Models (MRASM) are extended to handle multiple views of faces for finding facial landmarks automatically. In addition, temporal information is used while tracking in image sequences. The tracked facial landmark sequences are normalized and given as input to the expression classifier, which is based on the multivariate Continuous Hidden Markov Model.

The contribution of this thesis is the improvement of MRASM searching such that the landmarks are fitted in each frame using MRASMs for *multiple views* of faces, and the best fitted shape which is most similar to the shape found in the preceding frame is chosen. This way, *temporal information* is used for achieving consistency between consecutive frames. When the found shape is not trusted, deformation of the tracked shape is avoided by leaving that frame as empty and re-initializing the tracker. Afterwards, the empty frames are filled using interpolation, and α-trimmed mean filtering is performed on the landmark trajectories to eliminate the erroneous frames. Another contribution is that a complete non-manual gesture recognition system is developed, where the tracked landmarks are classified into one of the non-manual gestures based on CHMM classifier.

The proposed system is experimented on a database consisting of most common non-manual signs in Turkish Sign Language. In this database, we collected 5 repetitions from 11 (6 female, 5 male) different subjects for 8 different classes of non-manual signs. Due to the difficulty of collecting ground truth of all subjects by manually annotating landmarks in frame sequences, we selected 3 repetitions of 4 classes from 4 subjects (2 female, 2 male) and annotated them for further comparison.

Multi-view vs. single view and person specific vs. generic approaches are analyzed by implementing these variations of MRASM trackers and performing the classification using the tracking results of each approach. It is shown that the proposed technique of using temporal information while tracking increased the robustness of the system. Shape deformation is avoided throughout the frames and the tracker is able to catch the correct locations of landmarks by re-initialization if the found shape is not trusted. The system was able to attain best results with the proposed multi-view MRASM approach. The worst recognition rates found with this approach were 72.86% and 68.57% for person specific and generic approaches, respectively, when the training is done using three subjects, and leaving the remaining subject for testing. When the testing is done on an unseen video of a subject, whose other repetitions are involved in the training set, the accuracy was about 85%. It is seen that the tracker may be confused when distinguishing between classes with the same head movement but different facial expressions. But instead of including all classes, if the classes in question are decreased to cover only non-relevant classes, the system is able to recognize all, i.e. it works with 100% performance.

5.1. Remarks and Future Directions

The multi-resolution approach is seen as a crucial component of a facial landmark tracker since the active models are very sensitive to initialization and converge to local minima if only one resolution is used.

We used four different views, namely frontal, left, right, and upwards for extending MRASM approach for multiple views. All samples of the subjects are manually divided into four subsets concerning these views. A better way of doing this would be clustering the sample space into subsets using an unsupervised clustering method, such as k-means. This way, the clustering would divide the training set better and would not need to be done manually on training sets different than ours.

As discussed, person specific training and tracking performs considerably better than the generic approach. So, the usability of the application can be improved such that a new user would train the tracker for herself easily via a user-friendly interface. This way, we eliminate adding the user's samples to the database and training the tracker from scratch.

Last but not least, *working in real-time* is a significant property of an expression recognizer. Unfortunately, the implementation of the proposed system does not work in real-time and needs to be improved. Currently, each MRASM search takes about 3-4 seconds. Mature modules of the current MATLAB implementation should be optimized and written using C to increase speed. Another addition should be to use fewer landmarks for higher levels of MRASM and increase them by upsampling in the lower dimensions. We hope to make it run on real time. Additionally, the system should be adapted to work with a hand gesture recognizer concurrently which is needed to recognize whole sign language words instead of only recognizing the non-manual components.

REFERENCES

1. Ekman, P. and W. Friesen, *Facial Action Coding System*, Consulting Psychologists Press, 1978.

2. Fasel, B. and J. Luettin, “Automatic Facial Expression Analysis: A Survey”, *Pattern Recognition*, Vol. 36, No. 1, pp. 259–275, 2003.

3. Ong, S. and S. Ranganath, “Automatic Sign Llanguage Analysis: A Survey and the Future Beyond Lexical Meaning”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 27, No. 6, pp. 873–891, 2005.

4. Pantie, M. and L. Rothkrantz, “Automatic Analysis of Facial Expressions: the State of the Art”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 22, No. 12, pp. 1424–1445, 2000.

5. Spors, S. and R. Rabenstein, “A Real-time Face Tracker for Color Video”, *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP’01). 2001 IEEE International Conference on*, Vol. 3, 2001.

6. Jordao, L., M. Perrone, J. Costeira, and J. Santos-Victor, “Active Face and Feature Tracking”, *Proceedings of the 10th International Conference on Image Analysis and Processing*, pp. 572–577, 1999.

7. Ahlberg, J., “An Active Model for Facial Feature Tracking”, *EURASIP Journal on Applied Signal Processing*, Vol. 2002, No. 6, pp. 566–571, 2002.

8. Kapoor, A. and R. Picard, “A Real-time Head Nod and Shake Detector”, *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, pp. 1–5, 2001.

9. Zhang, Y. and Q. Ji, “Active and Dynamic Information Fusion for Facial Expression Understanding from Image Sequences”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 27, No. 5, pp. 699–714, 2005.

10. Inc, I., “The OpenCV Open Source Computer Vision Library”, `http://sourceforge.net/projects/opencvlibrary/`.

11. Viola, P. and M. Jones, “Rapid object detection using a boosted cascade of simple features”, *Proc. CVPR*, Vol. 1, pp. 511–518, 2001.

12. Fasel, I., R. Dahl, J. Hershey, B. Fortenberry, J. Susskind, and J. Movellan, “The machine perception toolbox”, 2004, `http://mplab.ucsd.edu/grants/project1/free-software/mptwebsite/API/index.html`.

13. Wei, X., Z. Zhu, L. Yin, and Q. Ji, “A Real Time Face Tracking And Animation System”, *Computer Vision and Pattern Recognition Workshop, 2004 Conference on*, pp. 71–71, 2004.

14. Dubuisson, S., F. Davoine, and J. Cocquerez, “Automatic Facial Feature Extraction and Facial Expression Recognition”, *3rd International Conference on Audio and Video Based Biometric Person Authentication*, 2001.

15. Lien, J., T. Kanade, and J. Cohn, “Automated Facial Expression Recognition Based on FACS Action Units”, *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 390–395, 1998.

16. Buciu, I. and I. Pitas, “Application of Non-negative and Local Non-negative Matrix Factorization to Facial Expression Recognition”, *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, Vol. 1, 2004.

17. Dubuisson, S., F. Davoine, and M. Masson, “A Solution for Facial Expression Representation and Recognition”, *Signal Processing: Image Communication*, Vol. 17,

No. 9, pp. 657–673, 2002.

18. Franco, L. and A. Treves, “A Neural Network Facial Expression Recognition System Using Unsupervised Local Processing”, *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, pp. 628–632, 2001.

19. Cootes, T., G. Edwards, and C. Taylor, “Active Appearance Models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 6, pp. 681–685, 2001.

20. Abboud, B. and F. Davoine, “Appearance Factorization Based Facial Expression Recognition and Synthesis”, *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, Vol. 4, 2004.

21. Cootes, T., C. Taylor, D. Cooper, J. Graham, *et al.*, “Active shape models-their training and application”, *Computer Vision and Image Understanding*, Vol. 61, No. 1, pp. 38–59, 1995.

22. Votsis, G., A. Drosopoulos, and S. Kollias, “A Modular Approach to Facial Feature Segmentation on Real Sequences”, *Signal Processing: Image Communication*, Vol. 18, No. 1, pp. 67–89, 2003.

23. Gokturk, S., J. Bouguet, and R. Grzeszczuk, “A Data-driven Model for Monocular Face Tracking”, *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Vol. 2, 2001.

24. Sebe, N., M. Lew, I. Cohen, Y. Sun, T. Gevers, and T. Huang, “Authentic Facial Expression Analysis”, *Int. Conf. on Automatic Face and Gesture Recognition*, 2004.

25. Cohen, I., *Automatic Facial Expression Recognition from Video Sequences Using Temporal Information*, Master’s thesis, University of Illinois at Urbana-

Champaign, 2000.

26. Sobottka, K. and I. Pitas, "A Fully Automatic Approach to Facial Feature Detection and Tracking", *Audio Visual Biometric Person Authentication*, 1997.

27. Terzopoulos, D. and K. Waters, "Analysis of Facial Images Using Physical and Anatomical Models", *Computer Vision, 1990. Proceedings, Third International Conference on*, pp. 727–732, 1990.

28. Tao, H. and T. Huang, "Connected vibrations: A modal analysis approach to non-rigid motion tracking", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 735–740, 1998.

29. Calder, A., A. Burton, P. Miller, A. Young, and S. Akamatsu, "A Principal Component Analysis of Facial Expressions", *Vision Research*, Vol. 41, No. 9, pp. 1179–1208, 2001.

30. Buciu, I., "A New Sparse Image Representation Algorithm Applied to Facial Expression Recognition", *Machine Learning for Signal Processing, 2004. Proceedings of the 2004 14th IEEE Signal Processing Society Workshop*, pp. 539–548, 2004.

31. Kanade, T. and J. Tian, "Comprehensive Database for Facial Expression Analysis", *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 46–53, 2000.

32. Lyons, M., S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding Facial Expressions with Gabor Wavelets", *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 200–205, 1998.

33. Moon, H. and P. Phillips, "The FERET verification testing protocol for face recognitionalgorithms", *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 48–53, 1998.

34. Ari, I., A. Uyar, and L. Akarun, "Facial Feature Tracking and Expression Recognition for Sign Language", *International Symposium on Computer and Information Sciences*, 2008, submitted.

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

36. Aran, O., I. Ari, A. Benoit, A. Carrillo, F. Fanard, P. Campr, L. Akarun, A. Caplier, M. Rombaut, and B. Sankur, "Sign Language Tutoring Tool", *Proceedings of eNTERFACE 2007, The Summer Workshop on Multimodal Interfaces*, Vol. 21, 2006.

37. Ma, J., W. Gao, J. Wu, and C. Wang, "A Continuous Chinese Sign Language Recognition System", *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 428–433, 2000.

38. Bowden, R., D. Windridge, T. Kadir, A. Zisserman, and M. Brady, "A Linguistic Feature Vector for the Visual Interpretation of Sign Language", *Proc. 8th European Conference on Computer Vision, Prague*, pp. 391–401, 2004.

39. Sagawa, H. and M. Takeuchi, "A Method for Recognizing a Sequence of Sign Language Words Represented in a Japanese Sign Language Sentence", *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 434–439, 2000.

40. Zhao, L., K. Kipper, W. Schuler, C. Vogler, N. Badler, and M. Palmer, "A Machine Translation System from English to American Sign Language", *Association for Machine Translation in the Americas*, 2000.

41. Keskin, C., A. Erkan, and L. Akarun, "Real Time Hand Tracking and 3D Gesture Recognition for Interactive Interfaces Using Hmm", *ICANN/ICONIPP*, 2003.

42. Aran, O., C. Keskin, and L. Akarun, “Sign Language Tutoring Tool”, *Proceedings of European Signal Processing Conference, EUSIPCO’05*, Vol. 5, 2005.

43. Aran, O., I. Ari, A. Guvensan, H. Haberdar, Z. Kurt, I. Turkmen, A. Uyar, and L. Akarun, “A Database of Non-Manual Signs in Turkish Sign Language”, *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th*, pp. 1–4, 2007.

44. Cootes, T. and C. Taylor, “Statistical models of appearance for computer vision”, *World Wide Web Publication, February*, 2001.

45. Stegmann, M. B., “Active Appearance Models: Theory, Extensions and Cases”, p. 262, 2000.

46. Stegmann, M. B., B. K. Ersboll, and R. Larsen, “FAME - A Flexible Appearance Modelling Environment”, *IEEE Transactions on Medical Imaging*, Vol. 22, No. 10, pp. 1319–1331, 2003.

47. Hamarneh, G. and T. Gustavsson, “Deformable spatio-temporal shape models: extending active shape models to 2D + time”, *Image and Vision Computing*, Vol. 22, pp. 461–470, 2004.

48. Mitchell, S., J. Bosch, B. Lelieveldt, R. van der Geest, J. Reiber, and M. Sonka, “3-D active appearance models: segmentation of cardiac MR and ultrasound images”, *Medical Imaging, IEEE Transactions on*, Vol. 21, No. 9, pp. 1167–1178, 2002.

49. Koschan, A., S. Kang, J. Paik, B. Abidi, and M. Abidi, “Color active shape models for tracking non-rigid objects”, *Pattern Recognition Letters*, Vol. 24, No. 11, pp. 1751–1765, 2003.

50. Pearson, K., “On lines and planes of closest fit to systems of points in space”, *Philosophical Magazine*, Vol. 2, No. 6, pp. 559–572, 1901.

51. Hotelling, H., “Analysis of a complex of statistical variables into principal components”, *Journal of Educational Psychology*, Vol. 24, No. 6, pp. 417–441, 1933.

52. Jolliffe, I., “Principal component analysis”, 1986.

53. Gross, R., I. Matthews, and S. Baker, “Generic vs. person specific active appearance models”, *Image and Vision Computing*, Vol. 23, No. 12, pp. 1080–1093, 2005.

54. Cootes, T., G. Wheeler, K. Walker, and C. Taylor, “View-based Active Appearance Models”, *Image and Vision Computing*, Vol. 20, No. 9-10, pp. 657–664, 2002.

55. Burges, C., “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, pp. 121–167, 1998.

56. Rabiner, L., “A tutorial on hidden Markov models and selected applications inspeech recognition”, *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, 1989.

57. Alpaydin, E., “Introduction To Machine Learning”, 2004.

58. Rabiner, L. and B. Juang, “Fundamentals of speech recognition”, 1993.

59. Jelinek, F., “Statistical Methods for Speech Recognition”, 1998.

60. Murphy, K., “Hidden Markov Model (HMM) Toolbox for Matlab”, `http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html`.