2-1-2009

Face recognition with variation in pose angle using face graphs

Sooraj Kumar

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Face Recognition with Variation in Pose Angle

Using Face Graphs

by

Sooraj Kumar

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

Supervised by

Dr. Andreas Savakis
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, NY
February 2009

Approved By:

____________________________________________________________________

Dr. Andreas Savakis, Professor and Department Head
Primary Advisor – R.I.T. Dept. of Computer Engineering

____________________________________________________________________

Dr. Muhammad Shaaban, Associate Professor
Secondary Advisor – R.I.T. Dept. of Computer Engineering

____________________________________________________________________

Dr. Juan Carlos Cockburn, Associate Professor
Secondary Advisor – R.I.T. Dept. of Computer Engineering
I would like to take this opportunity to thank my advisor, Dr. Andreas Savakis, who has mentored and guided me during the course of this thesis. I would also like to thank my thesis committee members, who have given me their support in making this research work possible, and my colleagues who have provided feedback at the numerous meetings throughout the duration of this work. The research in this thesis uses the CAS-PEAL-R1 face database collected under the sponsorship of the Chinese National Hi-Tech Program and ISVISION Tech. Co. Ltd.
Abstract

Automatic recognition of human faces is an important and growing field. Several real-world applications have started to rely on the accuracy of computer-based face recognition systems for their own performance in terms of efficiency, safety and reliability. Many algorithms have already been established in terms of frontal face recognition, where the person to be recognized is looking directly at the camera. More recently, methods for non-frontal face recognition have been proposed. These include work related to 3D rigid face models, component-based 3D morphable models, eigenfaces and elastic bunched graph matching (EBGM).

This thesis extends recognition algorithm based on EBGM to establish better face recognition across pose variation. Facial features are localized using active shape models and face recognition is based on elastic bunch graph matching. Recognition is performed by comparing feature descriptors based on Gabor wavelets for various orientations and scales, called jets. Two novel recognition schemes, feature weighting and jet-mapping, are proposed for improved performance of the base scheme, and a combination of the two schemes is considered as a further enhancement. The improvements in performance have been evaluated by studying recognition rates on an existing database and comparing the results with the base recognition scheme over which the schemes have been developed. Improvement of up to 20% has been observed for face pose variation as large as 45°.
# Table of Contents

**Chapter 1**  Introduction........................................................................................................ 1  

1.1.  *Face Detection and Recognition*.......................................................................... 1  

1.2.  *Challenges of Pose-invariant Face Recognition*............................................. 3  

1.3.  *Thesis Contribution*.......................................................................................... 4  

1.4.  *Thesis Outline* ................................................................................................. 5  

**Chapter 2**  Background ................................................................................................. 6  

2.1.  *Active Shape Models*......................................................................................... 6  

2.1.1  Point Distribution Model ............................................................................... 7  

2.1.2  Training the PDM Model.................................................................................. 8  

2.1.3  Initializing the ASM in the Image Frame....................................................... 9  

2.1.4  Calculating a Suggested Movement for Each Model Point............................ 9  

2.1.5  Using the PDM as the Local Optimizer for the Contour......................... 12  

2.1.6  Termination of Iteration under Suitable Condition................................... 15  

2.2.  *Gabor Wavelets, Jets and the Similarity Function* ..................................... 16  

2.3.  *Elastic Bunch Graph Matching* ...................................................................... 18  

2.3.1  Face Representation....................................................................................... 20  

2.3.2  Face Recognition........................................................................................... 25  

**Chapter 3**  Automatic Feature Point Positioning....................................................... 27  

3.1.  *Obtaining a priori Information*......................................................................... 28
3.2. *Automatic Placement Scheme* ............................................................... 29

3.2.1 Correcting Face Image Alignment .................................................... 30

3.2.2 Find the best-fit locations of fiducial points ..................................... 32

3.3. *Evaluation of the procedure* ............................................................. 37

Chapter 4  *Feature Weighting and Face Derotation* ......................... 40

4.1. *Face Recognition Using Elastic Bunch Graph Matching* ............... 41

4.2. *Feature Weighting Scheme* ............................................................. 42

4.2.1 Motivation ......................................................................................... 42

4.2.2 Procedure ........................................................................................ 43

4.2.3 Evaluation of Weighted Recognition Scheme ............................... 49

4.3. *Face Derotation Scheme* ................................................................. 54

4.3.1 Motivation ......................................................................................... 54

4.3.2 Procedure ........................................................................................ 56

4.3.3 Evaluation of Derotation Results ...................................................... 62

Chapter 5  *Pose-invariant Face Recognition with Jet-Mapping* .......... 66

5.1. *Motivation* ....................................................................................... 66

5.2. *Procedure* ....................................................................................... 67

5.2.1 Study of Jet Coefficients ................................................................. 67

5.2.2 Suggesting a Mapping ................................................................. 69

5.3. *Evaluation of Jet Mapping Scheme* ............................................... 71

5.4. *Combination of Jet Mapping and Weighting Schemes* ............... 75
5.5. Evaluation of the Combined Recognition Scheme .................................. 77

Chapter 6 Conclusions and Future Works ................................................. 83

6.1. Summary of works .................................................................................. 83
  6.1.1 Automatic Feature Point Positioning ................................................ 83
  6.1.2 Feature-Weighted Recognition Scheme .......................................... 83
  6.1.3 Face Derotation Scheme .................................................................. 83
  6.1.4 Jet-Mapped Recognition Scheme .................................................... 84
  6.1.5 A Combined Recognition Scheme ................................................... 85

6.2. Scope for Possible Future Work ........................................................... 86
  6.2.1 Training for Face Derotation ............................................................. 86
  6.2.2 Improving Automatic Feature Point Placement .............................. 86
  6.2.3 Real-Time Implementation ............................................................... 86

Bibliography ................................................................................................. 88
List of Figures

Figure 1 An example of a Shape Model. ................................................................. 7
Figure 2 Sampling Points along the Contour Normal........................................... 10
Figure 3 Finding the Best-Fit estimates along the Normal................................. 11
Figure 4 Applying Displacement Constraint ...................................................... 15
Figure 5 A 2D Gabor Wavelet ........................................................................ 16
Figure 6 A Model Graph .................................................................................. 19
Figure 7 A Bunch Graph ................................................................................... 20
Figure 8 Fiducial Points and the Face Graph structure ..................................... 21
Figure 9 Displacement Estimation .................................................................. 24
Figure 10 A Face Image with locations of features given as priori information .... 29
Figure 11 Block Diagram illustrating Automatic Feature Point Positioning .......... 30
Figure 12 3D Rotation components: Yaw, Pitch and Roll ................................. 31
Figure 13 Face Alignment Process ................................................................... 32
Figure 14 The Different ASM Shapes Used ...................................................... 33
Figure 15 Fitting the ASM to extract the required graph node positions ............ 34
Figure 16 Face Graphs Used for Different Face Poses ....................................... 35
Figure 17 Finding best fit points around the Eyes ............................................. 36
Figure 18 Finding Lip Corner By Finding the Steepest Edge in the neighborhood ... 36
Figure 19 Performance of Automatic Feature Point Positioning in comparison with

Manually Selected Nodes .................................................................................. 38
Figure 20 Recognition rates for Direct Recognition scheme .............................. 39
Figure 21 Block Diagram illustrating Direct Recognition Scheme ..................... 41
List of Tables

Table 1 Description of the Feature Points used in recognition........................................... 35
Table 2 Calculation of weights for 15°, 30° and 45° Face Pose Graph................................. 48
Table 3 Evaluation Results for Weighted Face Recognition scheme using Manually

    Positioned Feature Points for 15° Face Pose Query Images........................................ 51
Table 4 Evaluation Results for Weighted Face Recognition scheme using Manually

    Positioned Feature Points for 30° Face Pose Query Images........................................ 51
Table 5 Evaluation Results for Weighted Face Recognition scheme using Manually

    Positioned Feature Points for 45° Face Pose Query Images........................................ 50
Table 6 Evaluation Results for Weighted Face Recognition scheme using Automatic

    Feature Points Positioning for 15° Face Pose Query Images ...................................... 51
Table 7 Evaluation Results for Weighted Face Recognition scheme using Automatic

    Feature Points Positioning for 30° Face Pose Query Images ...................................... 52
Table 8 Evaluation Results for Weighted Face Recognition scheme using Automatic

    Feature Points Positioning for 45° Face Pose Query Images ...................................... 52
Table 9 Evaluation Results for Derotated Face Recognition scheme using Manual Feature

    Points Positioning for 22° Face Pose Query Images .................................................... 63
Table 10 Evaluation Results for Derotated Face Recognition scheme using Manual

    Feature Points Positioning for 22° Face Pose Query Images ....................................... 63
Table 11 Evaluation Results for Jet-Mapped Face Recognition scheme using Manually

    Positioned Feature Points for 15° Face Pose Query Images......................................... 72
Table 12 Evaluation Results for Jet-Mapped Face Recognition scheme using Manually

    Positioned Feature Points for 30° Face Pose Query Images......................................... 72
Table 13 Evaluation Results for Jet-Mapped Face Recognition scheme using Manually Positioned Feature Points for 45° Face Pose Query Images........................................... 73

Table 14 Evaluation Results for Jet-Mapped Face Recognition scheme using Automatic Feature Point Positioning for 15° Face Pose Query Images........................................... 73

Table 15 Evaluation Results for Jet-Mapped Face Recognition scheme using Automatic Feature Point Positioning for 30° Face Pose Query Images........................................... 73

Table 16 Evaluation Results for Jet-Mapped Face Recognition scheme using Automatic Feature Point Positioning for 45° Face Pose Query Images........................................... 74

Table 17 Calculation of Weights for Jet-Mapped Face Pose Graphs........................................... 78

Table 18 Evaluation Results for Jet-Mapped Face Recognition scheme using Manually Positioned Feature Points for 15° Face Pose Query Images........................................... 79

Table 19 Evaluation Results for Combined Recognition scheme using Manually Positioned Feature Points for 30° Face Pose Query Images........................................... 80

Table 20 Evaluation Results for Jet-Mapped Face Recognition scheme using Manually Positioned Feature Points for 45° Face Pose Query Images........................................... 80

Table 21 Evaluation Results for Jet-Mapped Face Recognition scheme using Automatic Feature Point Positioning for 15° Face Pose Query Images........................................... 80

Table 22 Evaluation Results for Combined Recognition scheme using Automatic Feature Point Positioning for 30° Face Pose Query Images........................................... 81

Table 23 Evaluation Results for Jet-Mapped Face Recognition scheme using Automatic Feature Point Positioning for 45° Face Pose Query Images........................................... 81
# Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition of the term as used in the thesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td>Active Contour Model</td>
</tr>
<tr>
<td>ASM</td>
<td>Active Shape Model</td>
</tr>
<tr>
<td>EBGM</td>
<td>Elastic Bunch Graph Matching</td>
</tr>
<tr>
<td>Face Library</td>
<td>A repository of frontal face images of known identities that are used for face recognition.</td>
</tr>
<tr>
<td>Model Face Image</td>
<td>A frontal face image in the face library.</td>
</tr>
<tr>
<td>Query Face Image</td>
<td>A face image without pose constraints that is to be recognized by matching with images into the face image library.</td>
</tr>
<tr>
<td>Face Profile</td>
<td>The vertical line that divides the face into two roughly symmetrical parts.</td>
</tr>
<tr>
<td>Face Profile Plane</td>
<td>The vertical plane passing through the vertical axis and the face profile line.</td>
</tr>
<tr>
<td>Pose Angle</td>
<td>Geometric angle made by the intersection of the face profile plane and the vertical plane normal to the camera image plane.</td>
</tr>
<tr>
<td>Face Detection</td>
<td>Finding a face in an image.</td>
</tr>
<tr>
<td>Face Recognition</td>
<td>Identifying the individual by comparing a given face to known, or trained, faces present in the Face Library.</td>
</tr>
<tr>
<td>Face Alignment</td>
<td>Adjusting the face features to a given orientation in the image plane, so as to correlate better with other images in the face library.</td>
</tr>
</tbody>
</table>
Chapter 1  Introduction

1.1. Face Detection and Recognition

Face Recognition is an important and ever-growing area of interest in the field of Computer Vision. This is a classification problem, where known faces of several individuals are ‘trained’ beforehand by feature extraction, and upon introduction of a new, unclassified face, the individual from the library of ‘trained’ individuals is to be identified.

Face recognition systems typically consist of two basic steps. The first step involves detecting the face position on the image frame and identifying facial features of interest. The latter step involves extracting the facial information and using it to recognize the face across a library of pre-trained faces. Both steps are quite elaborate, and hence, have been considered as separate problems to be solved independently. However, the performance of the face detection step directly affects the face recognition step. Hence, it is still important that both steps be performed with as much accuracy as possible.

Face detection has become quite a mature field of study. Rapid object detection using a boosted cascade of simple features [1], proposed by Viola and Jones, is one such face detection method. Other methods, such as Gravity-based Template Matching [2], achieve good face detection results, but lack in performance speed. Cascaded Harr classifiers [1], [3] also perform face and feature detection at a real-time rate. There are video-based face-tracking methods that make use of motion data that can be extracted by optical flow techniques [4]. Once face detection has been successful in finding a face
region, methods like Active Contour Models (ACMs) [5] are available for extracting local face feature regions based on standard face shapes.

The face recognition step can be chiefly divided into two broad categories: recognition by whole face, or recognition by components. Most of the basic principles behind face recognition can be developed from the understanding of human vision concepts [6]. Holistic face recognition methods, such as Principal Component Analysis-based eigenfaces [7] match the entire face.

Recognition by parts is implemented in approaches such as component-based face recognition with morphable models [8] and elastic bunched graph matching [9]. Modular eigenspaces [10] is an alternative approach to using eigenfaces. This methodology involves recognizing key components of the face individually and then piecing the recognized parts together to identify the face. Recognition by parts allows for a small amount of flexibility in the face structure, which is more robust.

Due to the three-dimensional structure of the human face, and the complexities involved in each of the different techniques used in face recognition solutions, slight changes of the face parameters in the query image, such as variation in face pose or expression, or changes in the parameters of the environment itself, such as variation in illumination, have significant impact on the quality of the recognition performed. Real-world implementations of face recognition systems have at least a few of these variations as challenges to overcome. Thus, making recognition algorithms robust with respect to one or more of these variable parameters is the focus of on-going research.
1.2. Challenges of Pose-invariant Face Recognition

The human face is a three-dimensional structure, and its two-dimensional projection varies dramatically with respect to the point/plane of view, i.e. variation in face pose with respect to the image capturing device. This is a cause of concern for any recognition algorithm that relies solely on basic image processing techniques to find similarities between two-dimensional images, when there is disparity in the points of view between the two face images.

Face images in training databases are usually restricted to a very small set of known face poses. Most training is done only on frontal face poses and, in some cases, full-profile face poses. These poses are generally easy to obtain from the subjects. However, in an unconstrained environment, which is the case in real-world implementations of face recognition algorithms, the query face images are usually not in one of the trained face poses. This disparity in face pose, between the query and trained images, is the cause of significant reduction in recognition rates.

Depending on the base algorithm used for recognition, steps can be implemented to eliminate recognition disparity due to pose variation. On the other hand, some recognition algorithms have been developed specifically to tackle the pose variation issue. Sanderson. et al [11] propose a statistical face model that can be used to generate a non-frontal pose face image from a trained frontal pose image. There has also been research making use of 3D face models. Borstein et al [12] generate a 3D model from a set of face images with small variations in pose or illuminations, and use the 3D model for recognition. However, the complexities of these algorithms are not suitable for real-time processing.
Gabor wavelets [13] are a prominent mathematical tool that can help describe features through a series of convolutions. Much research has been done using Gabor wavelet-based filters for frontal face recognition [14] [15], and some research pertaining to fixed-pose face recognition [9]. However, recognition across pose using Gabor wavelets has not been considered up to this point in time.

1.3. Thesis Contribution

This thesis considers the application of Gabor wavelets in pose-variant face recognition using Elastic Bunch Graph Matching as the base recognition system. The thesis then implements different independent procedures in trying to improve the face recognition rate of the base system across face pose variation and studies the robustness of each procedure towards face pose variation.

Considering the fact that the human face is almost symmetric along the vertical center, this thesis focuses only on pose variation along one direction, and assumes that a similar set of procedures can be applied for pose variation in the opposite direction as well. All experiments are carried out on faces gradually rotated in pose in an anti-clockwise direction.

The major contributions of this thesis are as bulleted below:

- An automated procedure for extracting essential feature points corresponding to facial features, using a few heuristics, is introduced and considered as a basis for face recognition.

- A novel approach to improving face recognition with the help of image mappings, called Face Derotation, is studied and the possibility of its application towards improving pose-variant face recognition is evaluated.
Another novel approach to improving pose-variant face recognition using Gabor jets, called Jet Mapping, is proposed and the performance results are studied.

1.4. Thesis Outline

The remaining chapters are documented as follows. Chapter 2 details previously developed methodologies that have been implemented in this work. Elastic Bunch Graph Matching, Active Shape Models, and the use of Gabor wavelets for recognition are explained. Chapter 3 introduces a novel process for acquiring feature points from the face image, and describes the integration of existing techniques used to get best possible results. Chapter 4 introduces some techniques used for improving face recognition performance across small variation in pose, which are implemented after the automatic feature extraction process. The processes covered here include face derotation, and weighted similarity matching of feature nodes. Chapter 5 introduces a novel procedure for improving recognition of faces across pose, called Jet Mapping. This procedure introduces a trained mapping stage between the feature extraction and recognition stages. The training and mapping processes are explained here. Also, an extension to this procedure, implementing the concept of weighted similarity matching is explained. The performances of the procedures explained here are evaluated along with those mentioned in the previous chapters. Finally, Chapter 6 provides a concluding summary of the various implementations introduced in this thesis, and comments for future research work.
Chapter 2  Background

This chapter reviews previous methods that are being implemented in this thesis work. Section 2.1 explains the concept of Active Shape Models, which is relevant in finding object contours in images. Active Shape Models are used later in Chapter 3 to support the Automatic Feature Point Positioning scheme. Section 2.2 covers the basics involved in performing recognition using Gabor wavelets and jets. This is further used throughout this thesis as a basis for recognition, and is modified as needed in each scheme that is introduced in Chapters 4 and 5. Section 2.3 also covers the implementation of a face recognition scheme called Elastic Bunch Graph Matching, which uses Gabor wavelets and jets, described in Section 2.2, as its basis for recognition.

2.1. Active Shape Models

Active Shape Model (ASM) is a common approach used for finding and describing instances of an object of a given class, which can be represented by a deformation of a shape generic to that class of objects. ASMs are based on the concept of Active Contour Models [5], but have an added shape constraint to restrict the shape of the contour. ACMs are contours represented by chains of ‘nodes’, called snakes, which can iteratively stretch and deform to fit image features to which they are attracted, while simple local shape constrains are applied. ASMs [16] incorporate the concept of Point Distribution Models (PDMs) with the iterative scheme of ACMs, to generate and restrict the shape of the models accurately within the class of objects, and prevent implausible shapes from occurring.
2.1.1 Point Distribution Model

The Point Distribution Model (PDM) [17] helps generate a statistical representation of the contour points of shapes that fit different objects of the same class. This is based on the assumption that all of the shapes being represented have a flexible shape model of labeled nodes, over which controlled deformations can be applied to generate each individual shape, within a small margin of error. The necessary equations used in this thesis are reproduced in this section.

\[
x = \bar{x} + P b
\]

where \( x \) is the vector representing the positions of the \( n \) points in the shape, and

\[
x = (x_0, y_0, x_1, y_1, \ldots x_k, y_k, \ldots x_{n-1}, y_{n-1})^T
\]

\((x_k, y_k)\) represents the position of point \( k \) in the shape, \( \bar{x} \) is a vector representing the mean position of all the points, \( P = [p_1 \ p_2 \ \ldots \ p_t] \) is a matrix of the first \( t \) modes of variation, \( p_i \), corresponding to the most significant eigenvectors in a Principal
Component decomposition of the position variables, and \( \mathbf{b} = (b_1 \ b_2 \ \ldots \ b_t)^T \) is a vector of weights for each of the \( t \) modes.

The columns of \( \mathbf{P} \) are orthogonal. So

\[
\mathbf{P}^T \mathbf{P} = \mathbf{I}
\]

and from Equation (2.1):

\[
\mathbf{b} = \mathbf{P}^T (\mathbf{x} - \bar{\mathbf{x}})
\]  

(2.2)

The mean shape \( \bar{\mathbf{x}} \) and the modes of variation \( \mathbf{P} \) are calculated from a set of training examples. Equation (2.2) can be used to estimate the weights \( \mathbf{b} \) needed to generate a shape from the model that can best fit a given shape \( \mathbf{x} \).

Equation (2.1) allows the generation of new shapes from the class of shapes, by varying the parameters \( (b_i) \) within suitable limits. We can define the shape of a model object by just choosing the values of \( \mathbf{b} \). An instance \( \mathbf{X} \) of the model can be placed in an image frame, by defining a scale, orientation and position transform for the model, as in the following equation:

\[
\mathbf{X} = M(s, \theta)[\mathbf{x}] + \mathbf{X}_c
\]  

(2.3)

where \( \mathbf{X}_c = (X_c \ Y_c \ X_c \ Y_c \ \ldots \ X_c \ Y_c)^T \), \( M(s, \theta)[\cdot] \) is an operation defining rotation by \( \theta \) and scaling by \( s \), and \( (X_c,Y_c) \) is the new center of the model in the image frame coordinates.

2.1.2 Training the PDM Model

Training the PDM model from a training set of object models requires that the training set must first be standardized in orientation and scale of all the shape models, such that there is minimum possible displacement between all the corresponding labeled points in
the training set. Cootes et al [17] outline the process of standardizing a given set of shape models in order to obtain the best PDM model.

### 2.1.3 Initializing the ASM in the Image Frame

The pose and shape parameters of the ASM have to be initialized before any of the computations can be performed. Normally, the pose parameters would have to be initialized to arbitrary values, such that the ASM model is placed within a close proximity of the object in the image, so that the ASM can converge to the object contour. Wei Wang et al [18] propose using salient features, which can be easily located with good accuracy using automated detectors, to initialize the shape model and provide region constraints on the subsequent iterative shape searching.

### 2.1.4 Calculating a Suggested Movement for Each Model Point

Given an initial estimate of the positions of the active contour points, the next step is to evaluate the best fit positions of the points within their local proximity and generate a displacement vector, \( dX \), such that the boundary points move towards the edges of the image object that we are attempting to fit. There are several approaches to find this estimated adjustment. One approach is to move toward the strongest edge along a normal to the model boundary, as described in [19]. Another approach, which can be used in a broader context where contrasting boundaries may not constitute a part of the active shape contour, is to find the best local texture model match along the normal to the model boundary, as proposed by Wang et al [18].
2.1.4.1 Edge Constraint in Local Texture Model Matching

Local Texture Model Matching, as proposed by [18], makes use of the Mahalanobis distance function to estimate the best texture match position at each model point. However, this thesis shall utilize a simple least distance approach for finding the best texture match position along the normal of each point, as described below.

The texture model at a point in the shape model is the one-dimensional array of grayscale texture value sampled at $2K + 1$ points along the normal to the contour, with $K$ points extending on either side of the contour shape, as illustrated in Figure 2.

![Figure 2. Sampling Points Along the Contour Normal.](image)

A texture model $g_{ij}$, which is a vector of $2K + 1$ grayscale samples, is generated for the $i^{th}$ contour point in the $j^{th}$ training model. The set of texture models $\{g_{ij}\}$ at all the corresponding labeled points are averaged over the training set to get an ‘averaged’ texture model $\{\bar{g}_i\}$ for each contour point. This averaged texture model is then used in
estimating the best texture match positions in calculating a suggested movement for each model point in an iteration.

Figure 3. Finding the best fit estimates along the normal of each shape model node.
To find the best texture match position along a normal to a model point, we first have to sample \( M \) points along the contour points’ normal on either side of the contour, totaling to \( 2M + 1 \) grayscale samples, where \( M > K \). For the \( s^{th} \) sampling point along the normal, \(-L \leq s \leq L; L = (M - K)\), a window of \( 2K + 1 \) texture samples is taken from the sampled set of \( 2M + 1 \) values, centered at the \( s^{th} \) sampling point, as illustrated in Figure 3, and the distance \( d_{is} \) between the texture model \( g_{is} \) of the current window and the average node texture model \( \bar{g}_i \) is calculated.

\[
d_{is} = \sum_{p=-K}^{K} |g_{i(s-p)} - \bar{g}_{i(s-p)}|, \quad -(M - K) \leq s \leq (M - K)
\]

This is done for all the \( 2(M - K) + 1 \) sampling points along the normal of each model point \( i \). Then, the sample point \( s \) with the minimum texture distance \( d_{is} \) for that model point is considered to be the best fit position and the displacement to that sample point from the current model point position is taken as the suggested movement \((dX_i, dY_i)\) for that model point. The change in position of all the node points in the contour with respect to the image frame, calculated during an iteration, is represented by a vector \( dX \), where

\[
dX = (dX_0, dY_0, dX_1, dY_1 \ldots dX_{n-1}, dY_{n-1})^T.
\]

### 2.1.5 Using the PDM as the Local Optimizer for the Contour

ASM makes use of the PDM of an object as a constraint on the final deformation of the active contour at the end of iteration. Suppose the change in position of each of the node points in the contour, calculated during an iteration, is represented by a vector \( dX \), where

\[
dX = (dX_0, dY_0, dX_1, dY_1 \ldots dX_{n-1}, dY_{n-1})^T
\]
The current locations of the points in the image frame, represented by \( X \), have to be moved to their new positions \( X + dX \). This is brought about by a controlled change in the pose and shape parameters used for generating \( X \) from the PDM model.

If the current estimate of the model centered at \((X_C,Y_C)\), with scale \( s \) and orientation \( \theta \), new values of these pose parameters have to be estimated to better fit the image with the current model. This is done by finding the translation \((dX_C,dY_C)\), rotation \( d\theta \) and scaling factor \((1 + ds)\) from \( dX \) according to the procedure outlined in [19]. Once the pose variables are adjusted, the remaining residual adjustment in the model shape \( x \) is denoted as \( dx \). The residual adjustment needed \( dx \), has to be calculated in the object co-ordinate frame, so that:

\[
X + dX = M(s(1 + ds),\theta + d\theta)[x + dx] + (X_C + dX_C)
\]

Equation (2.4) includes coordinates expressed in two different coordinate frames. The object model \( X \) and the change \( dX \) are relative to the image coordinate frame, while the local object model \( x \) and the local deformation \( dx \) are relative to the object coordinate frame. Solving for \( dx \) gives the following equation:

\[
dx = M \left( (s(1 + ds))^{-1}, - (\theta + d\theta) \right) [M(s,\theta)[x] + dX - dX_C] - x
\]

This is the deformation required in the shape generated from the PDM using Equation (2.1), and can be satisfied by representing it as a change \( db \) in parameter \( b \).

\[
x + dx \approx \bar{x} + P(b + db)
\]

Since, there are only \( t \) modes of variation available in the model of Equation (2.1), and \( dx \) can move the points in \( 2n \) different degrees of freedom, we can achieve an approximation of the required deformation by allowing deformation in the \( t \) most
significant modes of variation observed in the training set. Such restriction enforces the global shape constraints. From Equations (2.1) and (2.6), we get:

\[ dx \approx P(db) \]

or

\[ db = P^T dx \quad (2.7) \]

Finally, these updated pose and shape parameters are used in the next iteration to generate a new starting model \( X \). This process is iterated until a suitable terminating condition is met.

### 2.1.5.1 Applying Local Constraints to Known Salient Features

Generally, it is possible that the ASM might not converge properly, even if satisfactory initialization is given. Such errors might not always be overcome by training more ASM models. However, errors can be reduced by placing certain constraints in each iteration step. This section describes the process of using known information to reduce the mentioned errors.

For example, if the ASM initializations are done with the help of known positions of features like pupil, mouth etc., then these features can be used again as positional constraints for the corresponding features of the ASM, to moderate the shape displacement at the end of an iteration before updating the model parameters [18].

If \( X \) is the shape vector in the image coordinate frame, \( dX \) is the displacement vector at the end of an iteration, \( P'_{k} = (X'_{k}, Y'_{k}) \) is the initialization location of the \( k^{th} \) point in the shape model \( X \), then we want to constrain the shift of the nodes around \( (X_{k}, Y_{k}) \) to be such that the node cluster stays close to the initial position \( P'_{k} \). If \( m = 1, \ldots M \) are the indices of the nodes for which the center of gravity is approximately at \( k^{th} \)
node, then the displacement for this cluster of $M$ nodes is modified by shifting the new gravity center, towards the original position, as given during the initialization.

![Center of Gravity](image)

(a) Before Correction  (b) After Correction

**Figure 4.** Applying displacement constraint on the center of gravity of feature nodes.

The required displacement, necessary to enforce the constraints, is calculated as the difference in position of the new gravity center of the $M$ nodes and the initial position $P_{k}$. This is done for each independent cluster of nodes with suitable initial positions.

### 2.1.6 Termination of Iteration under Suitable Condition

At the end of each ASM search iteration the suggested movement is examined. If a model point has its best fit positions close to the center of the normal, it is said to have converged. The suggested movement for the converged node, would be set to a zero vector. If a majority of the model nodes have converged, then it is assumed that the ASM has converged to its best fit position, and the iteration is stopped.

However, if the ASM has not yet converged to the object contour, the suggested movement would still be significant. In some situations, where the ASM might not have been able to converge to the object, either because it has been poorly initialized or because it could not track the contour properly, the suggested movement might not
converge to a zero vector, even after several iterations. In these cases, an iteration limit has to be placed to detect divergence from preferred object fit.

### 2.2. Gabor Wavelets, Jets and the Similarity Function

This section describes the mathematics behind Gabor wavelets, their use in constructing a jet, and their contribution towards face recognition. Gabor wavelets are biologically motivated convolution kernels and they exhibit desirable characteristics of spatial locality and orientation selectivity. As a result, the Gabor transformed face images produce salient local and discriminating features that are suitable for face recognition. The representation of local features in this work is based on the Gabor wavelet transform.

![Figure 5. A 2D Gabor Wavelet.](image)

The kernel of the Gabor wavelet consists of two components: the plane wave, with a constant frequency, orientation and amplitude, and a Gaussian envelope that restricts it. The generalized equation of a 2D Gabor wavelet kernel is given below.

\[
\psi_{\mu,\nu}(\vec{x}) = \frac{\|\vec{k}_{\mu,\nu}\|^2}{\sigma^2} e^{-\frac{\|\vec{k}_{\mu,\nu}\|^2}{2\sigma^2} \frac{||\vec{x}||^2}{2}} \left[ e^{ik_{\mu,\nu}\vec{x} - e^{-\frac{\sigma^2}{2}}} \right] \tag{2.8}
\]

where \(\mu\) and \(\nu\) define the orientation and scale of the Gabor kernels, \(\vec{x} = (x,y)\), \(||.||\) denotes the norm operator, and the wave vector \(k_{\mu,\nu}\) is defined as follows:
\[ \overrightarrow{k_{\mu,\nu}} = \begin{pmatrix} k_{\nu} \cos \varphi_{\mu} \\ k_{\nu} \sin \varphi_{\mu} \end{pmatrix}, \quad k_{\nu} = 2^{-\frac{\nu+2}{2}} \pi, \quad \varphi_{\mu} = \mu \frac{\pi}{8} \]  

(2.9)

A set of convolutions for kernels of different orientations and frequencies at one image pixel is called a jet. A jet describes a small patch of a grayscale image \( I(\vec{x}') \), around a given pixel \( \vec{x} = (x, y) \). It is based on the Gabor wavelet transform, which is defined as the following convolution.

\[ I(\vec{x}) \ast \psi_{\mu,\nu}(\vec{x}) = \int I(\vec{x}') \psi_{\mu,\nu}(\vec{x} - \vec{x}') d^2 \vec{x}' \]  

(2.10)

where * denotes convolution. The jet used in this work uses a set of Gabor wavelets, from Equation (2.8), that covers eight orientations, \( \mu = 0, \ldots 7 \) and 5 frequency scales, \( \nu = 0, \ldots 4 \). We use a coefficient index \( j = \mu + 8\nu \) to index the 40 different coefficients in the jet, with \( j = 0, \ldots 39 \). This sampling evenly covers a band in frequency space. The coefficients \( J_j(\vec{x}) \) of the jet \( J \) are defined with respect to equation (2.10) as follows:

\[ J_j(\vec{x}) = \int I(\vec{x}') \psi_{\mu,\nu}(\vec{x} - \vec{x}') d^2 \vec{x}' \]  

(2.11)

The second term in the brackets of Equation (2.8) makes the Gabor wavelet kernels DC-free. Gabor wavelets are robust as a data representation because of their biological relevance. Since they are DC-free, they also provide robustness against varying brightness in the image. The limited localization in space and frequency provides some robustness against translation, distortion, rotation and scaling. Only the phase changes drastically with translation, but can safely be ignored.

Recognition using Gabor jets is done with the help of similarity functions. Due to phase rotation, jets taken from the image points only a few pixels apart from each other have very different coefficients, although representing almost the same local feature. This can cause severe matching problems. Hence, the phase can either be ignored, or
compensated for its variation explicitly. The similarity function used in this work for comparing two jets, $\mathcal{J}$ and $\mathcal{J}'$, is

$$S_a(\mathcal{J}, \mathcal{J}') = \frac{\sum_j a_j a_j'}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}$$  (2.12)

This function ignores phase, and has been used extensively in previous works [20]. With a jet $\mathcal{J}$ taken from a fixed image position and jets $\mathcal{J}' = \mathcal{J}'(\vec{x})$ taken at a variable position $\vec{x}$, $S_a(\mathcal{J}, \mathcal{J}'(\vec{x}))$ is a smooth function with local optima. Equation (2.12) builds the basis of using jets of Gabor wavelet-convoluted coefficients for recognition purposes, and is used in all other recognition concepts.

### 2.3. Elastic Bunch Graph Matching

This section covers details on Elastic Bunch Graph Matching, which is a recognition scheme that builds a face model using Gabor wavelet-based jets as a feature descriptor. The basic recognition scheme introduced in this section is considered as the base evaluation scheme for the procedures introduced in Chapters 4 and 5. The first sub-section describes how face models are represented, and the second sub-section describes how the actual recognition process is carried out.

Elastic Bunch Graph Matching (EBGM) is a generic object recognition procedure that has been applied to the class of faces [20]. The basic face object representation is a labeled graph, which is the structural representation of a set of Gabor wavelet-based jets at fiducial points, such as the pupils, corners of mouth, etc., that are spatially constrained by loosely defined edges with neighboring jets. This general structure is useful for handling any kind of coherent object class, and may be sufficient for discriminating between structurally different object types.
Figure 6. A Model Graph structure representing nodes and edges.

A model graph, as implemented by Wiskott et al [9], uses jets generated from Gabor-wavelet based convolution for the representation of local features at the nodes of the graph structure. Wiskott et al also incorporate distance information along the graph edges. When the labeled graph, along with its jet information, is stored into a library, it is called a model graph. When a new labeled graph is generated from a new image, it is called an image graph. Image graphs can be stored into the library to become model graphs, or be directly used for recognition against the existing model graph library.

Bunch graphs are a collective representation of model graphs of several different subjects. Bunch graphs contain all the jets of the set of the faces it represents at every node. At each of these nodes, any one of the jets from any face in the set can be selected in combination with the jets from the other nodes in the bunch graph. In the case of pose-variant face recognition, bunch graphs are created for each set of faces in the same pose.
Figure 7. A Bunch Graph stores jet bunches for each feature point and the average edge distances between them from all the faces.

2.3.1 Face Representation

This thesis implements a subset of the face graph implemented by Wiskott et al [20]. This work uses twelve fiducial points in the face that can be selected conveniently even with large face pose variation. Due to the complex 3D structure of the human face, fiducial points on the side of the face that turns away from the camera plane, as the pose changes, become harder to select. If the face pose angle is large enough, some of these points may not even be visible in the image frame. Hence, the set of feature points being used for pose-variant face recognition would have to predominantly exist on the side of the face that directly faces the camera. For this reason, the feature point set used in this work lies only on one side of the face, as marked in Figure 8.
Figure 8. Fiducial points used from only one side of the face, connected by a face graph structure.

A labeled graph $G$ represents a face consisting of $N$ nodes on the fiducial points at positions $\vec{x}_n$, $n = 1, ..., N$. The nodes are labeled with jets $J^n$. The edges are labeled with distances $\Delta \vec{x}_e = \vec{x}_n - \vec{x}_{n'}$, $e = 1, ..., E$, where edge $e$ connects node $n'$ with $n$. Hence the edges are two-dimensional vectors. A labeled graph without the jet information is called a grid. The graph structure used in this work is maintained across pose variation, in terms of correspondence between feature points across pose. However, since there is significant variation in geometry and local features, bunch graphs are separately created for each different face pose trained.

A set of jets, referring to one fiducial point in the graph, is called a bunch. In the bunch graphs generated, each node would have a bunch of jets corresponding to that node from all the faces used to generate that bunch graph. While locating fiducial points on a new face, using procedures described in later sections, the algorithm tries to select the
best fitting jet from a bunch for each fiducial point. This is done so as to cover a much larger range of facial variation than represented in the trained models.

If, for a particular pose, there are $M$ model graphs $G_{m}^{B}, m = 1, \ldots, M$, from which a face bunch graph (FBG) is generated. The resultant face bunch graph would have the same structure, with its nodes linked with bunches of jets $J^{n,B_{m}}, n = 1, \ldots, N$, and its edges linked with the average distances of the corresponding edges, \( \Delta \bar{x}_{e}^{B} = \frac{1}{M} \sum_{m} \Delta \bar{x}_{e}^{B_{m}}, e = 1, \ldots, E \).

2.3.1.1 Manual Definition of Graphs during Training

For initial training of the EBGM, the face grid is manually fit by selecting the individual positions of the node points by visual judgment for their best fit positions. Once the nodes in the grid are manually placed, the edges between the nodes are charted out, and the edge information is obtained from the difference of the node positions. Finally, the jets at the nodes are obtained from the image to create the model graph.

2.3.1.2 Automatic Fiducial Point Placement

Once the initial training for a face graph of a pose is manually carried out for a well-populated set of faces and an FBG has been generated from the trained model graphs, an automatic procedure is capable of correcting node placements within a small displacement $\bar{d}$ in the order of a few pixels. This procedure is used for finding and positioning nodes at the nearest best-fit positions for the corresponding fiducial points, when a generic face graph of the concerned pose is placed over a new face for which an image graph is to be automatically extracted.
This work implements only a portion of the entire automation algorithm proposed by Wiskott et al, which deals with positioning the individual nodes after the entire face grid is placed over the face. Due to the limitation of the Gabor wavelets, it is assumed that the node points are within an error of eight pixels from the manually preferred best-fit node positions prior to this procedure, so as to get best estimates. The necessary equations were developed by Wiskott et al [9] and are reproduced below for convenience.

The phase information available in the jets can be used as a means for jet localization in the image, since it varies so quickly with change in location. Assuming that two jets $\mathcal{J}$ and $\mathcal{J}'$ refer to object locations with a small displacement $\vec{d}$, the phase shift on the wave vector $\vec{k}_{\mu,v}$ can be approximated as $\vec{d}\vec{k}_{\mu,v}$. Equation (2.12) for jet similarity, can be made phase-sensitive, and re-written below.

$$S_\phi (\mathcal{J}, \mathcal{J}') = \frac{\sum_j a_j a_j' \cos(\phi_j - \phi_j' - \vec{d}\vec{k}_{j})}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}$$

Equation (2.13)

The above equation reaches a maximum as the geometric distance between the two locations, which relate to the jets $\mathcal{J}$ and $\mathcal{J}'$, gets closer to zero. Thus, maximizing the equation (2.13), will give the displacement estimate $\vec{d}$. Maximizing the similarity function $S_\phi (\cdot)$ is done in its Taylor expansion form:

$$S_\phi (\mathcal{J}, \mathcal{J}') \approx \frac{\sum_j a_j a_j' \left[ 1 - 0.5 (\phi_j - \phi_j' - \vec{d}\vec{k}_j)^2 \right]}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}$$

Equation (2.14)

Solving for $\vec{d}$ in the above equation gives the solution:

$$\vec{d}(\mathcal{J}, \mathcal{J}') = \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \frac{1}{\Gamma_{xx} \Gamma_{yy} - \Gamma_{xy} \Gamma_{yx}} \times \begin{pmatrix} \Gamma_{yy} & -\Gamma_{yx} \\ -\Gamma_{xy} & \Gamma_{xx} \end{pmatrix} \begin{pmatrix} \Phi_x \\ \Phi_y \end{pmatrix}$$

Equation (2.15)

where $\Gamma_{xx} \Gamma_{yy} - \Gamma_{xy} \Gamma_{yx} \neq 0$ and
\[ \Phi_x = \sum_j a_j a_j' k_{jx} \left( \phi_j - \phi_j' \right), \quad \Gamma_{xy} = \sum_j a_j a_j' k_{jx} k_{jy} \]

and \( \Phi_y, \Gamma_{yx}, \Gamma_{xx}, \Gamma_{yy} \) are defined accordingly. In all the above equations, the phase difference has to be corrected to be within the range of \( \pm \pi \). The distance estimate obtained in Equation (2.15) gives the best-guess displacement to the local best-fit location for the concerned node jet, which can be applied to the current node position as a correction.

The following graph plots out the displacement estimation for an arbitrary point in a sample face image along its horizontal. It can be inferred that the displacement estimation function is accurate within a few pixels of displacement from its expected best fit position. As seen from the graph, this is a \( \pm 6 \) pixel window around the best fit point.

![Graph showing displacement estimation](image)

**Figure 9.** Displacement Estimation: (a) Similarity without phase (shown in dashed line), (b) Similarity with phase (shown in dotted line), and (c) Displacement estimation scaled down by a factor of 6.0 (shown in solid line).
2.3.2 Face Recognition

Recognition is carried out by use of the jet similarity functions. The similarity function, without phase, as represented in Equation (2.12) is used, since it would be more consistent across small variations, such as expression, etc., for the face of the same individual over different images. For recognition to be performed, model graphs of a set of faces have to be trained into the library, and an image graph from a query face image has to be obtained, either by manual placement or automatic placement procedure. The image graph is then compared against each model graph in the library, using a graph similarity function, and the comparison with the highest graph similarity is chosen as the matched pair.

There can be two distinct types of recognition that might need to be carried out. The first case is when both the model graph and image graph belong to the same face pose. This is the most direct approach, where all the jets have proper correspondence with their counterparts in either graphs. For an image graph $G^I$ and a model graph $G^M$, the graph similarity function is defined as:

$$S_g(G^I, G^M) = \frac{1}{N} \sum_{n=1}^{N} S_a(J^{n,I}, J^{n,M})$$

(2.16)

where $N$ is the total number of nodes in the graph. The second case of recognition is when recognition is being done across different graphs. In this case, the above function is modified so as to compute the average similarity of only the nodes that have correspondence between the two graphs. If $K$ nodes have correspondences between the graphs, and the node $p_k$ in the image graph corresponded to node $q_k$, $k = 1, ..., K$, then the above equation can be modified as:
\[ S_g(\mathcal{G}^I, \mathcal{G}^M) = \frac{1}{K} \sum_{k} S_a(\mathcal{J}^{p_{k,I}}, \mathcal{J}^{q_{k,M}}) \] (2.17)

However, in this thesis, the graphs across the various trained poses have been confined to a very small set of fiducial points such that the graphs have corresponding indices for all the nodes, so Equation (2.16) will be used for all cases of face recognition.

This section described a face recognition scheme that has been previously implemented as a part of Elastic Bunch Graph Matching using Gabor wavelet-based jets of facial features for recognition across pose. However, this previously established scheme has not considered the deformation of facial features over variation in face pose as an issue that has to be tackled to improve recognition rates across pose. The focus of this thesis is to improve the recognition rates over face pose variation, achieved by the base scheme described in this section. Chapters 4 and 5 introduce approaches developed by this thesis to improve the said recognition rates. Chapter 3 deals with the issue of placing the facial feature points accurately enough to be able to support the works in Chapters 4 and 5.
Chapter 3  Automatic Feature Point Positioning

This chapter introduces a proposed feature point placement scheme that implements a combination of Active Shape Models and Elastic Bunch Graph Matching, which have been discussed in Chapter 2 as background information.

Multi-pose face recognition is a direct extension of frontal face recognition. However, recognition across pose, or pose-variant face recognition, uses a query image with a different face pose for which an image does not exist in the library. In either situation, there are two generalized approaches to face recognition: model-based and appearance-based. Model-based approaches try to build a 2D or 3D model of the face, with feature descriptors that represent different segments of the face model, and make use of this model during recognition. Appearance-based approaches use direct pixel information from the image to perform the recognition.

This thesis uses a model-based face recognition approach and it extracts face features that are crucial to the recognition performance. There are several automated algorithms that detect faces in an image frame. However, face detection alone is not sufficient for face recognition. It is also essential that the features selected for recognition align well with the face, so that the individual feature descriptors used for recognition provide the best match results against the feature descriptors present in the library.

Variation in face pose creates the following two problems for local face features:
(a) Face features points can be obstructed in some face pose angles, preventing matching of all features across poses.
(b) Face features may become obscured over large pose variation, especially in feature-rich regions of the face, such as eyes, nose and mouth.
The selection of feature points used for face recognition across pose variation should consider these cases, so that a good set of features are represented. Furthermore, locating these features in the image frame should be easy to automate, so that manual intervention can be avoided. This chapter describes the implementation for automatically selecting the best fit locations of the feature points used in face recognition, while the following chapters will describe the importance of these feature nodes with respect to the face recognition.

Section 3.1 describes a priori information that is required for the proposed automatic feature point placement scheme and reasons the need for it. Section 3.2 describes the actual procedure that is proposed in this chapter, while the final section in this chapter performs an evaluation of the proposed procedure.

3.1. Obtaining a priori Information

This thesis implements a combination of Active Shape Models (ASMs) and Elastic Bunch Graph Matching (EBGM) to track the positions of the features of interest on a face at any pre-trained pose. A good initialization of feature points is crucial to getting good placement results for any search algorithm. The effects of poor initialization for an ASM search algorithm can be

(i) prolonged convergence time, and/or

(ii) convergence to non-face local matches.

Initialization is, thus, important for improving proper match results. The works of [18] implement the knowledge of features that are easily detectable with good accuracy to confine the search area for the ASM. This dramatically improves the fitting accuracy and reduces the number of search iterations needed to get convergence.
For initializing the ASM search space by confining feature positions, this thesis work currently obtains the location information of the eyes and mouth manually through interactive input. However, these selected features (eyes and mouth) can be detected through a series of well-trained Harr-cascade filters [3]. This would be an inexpensive alternative for finding the eyes and mouth features automatically without manual feedback.

Another parameter that needs to be given as prior knowledge is the face pose angle. Face pose estimation is a separate area of research and is outside the scope of this work. This thesis assumes that the location information of these three facial features (left eye, right eye, and mouth) and the face pose parameter can be obtained from \textit{a priori} steps dealing with pose estimation [21].

![Face Image with locations of features given as priori information.](image)

\textbf{Figure 10.} A Face Image with locations of features given as priori information.

\subsection*{3.2. Automatic Placement Scheme}

This section describes the actual process for automatically placing the feature points on the face image, after having collected the necessary \textit{a priori} information described in the previous section.
Knowing the positions of a few key feature points in the image frame allows the estimation of the scale and orientation of the face in the image frame. However, the most important step for proper face recognition results would be the alignment of the face image with the face images in the library. Although Gabor-wavelet based jets can withstand small variations in image orientation, it is still important that this variation is removed, since this thesis work tries to measure recognition improvement along other parameters. Section 3.2.1 describes this procedure.

After the face image is aligned, the next step is to select an ASM that corresponds to the face pose parameter given as *a priori* information, and fit it along the feature points used for face alignment, extract the points necessary for face recognition, and use a combination of heuristics and jet-displacement estimations to best-fit the positions of these points over the fiducial points of the face. Section 3.2.2 describes this procedure.

![Block Diagram illustrating Automatic Feature Point Positioning](image.png)

**Figure 11.** Block Diagram illustrating Automatic Feature Point Positioning.

### 3.2.1 Correcting Face Image Alignment

There are several variables involved when considering face pose. Any 3-dimensional rotation can be separated as a combination of rotations along three different axes. These
rotations are generally called yaw, pitch and roll, or can be collectively described as Euler angles.

![Figure 12. Rotation components: Yaw, Pitch and Roll.](image)

This thesis tries to study the impact on certain algorithms by variation of only one of these 3D rotation parameters – Yaw. For this, it is necessary to remove all other rotation parameters as much as possible. The simplest way to do this on a 2D face image would be to perform an affine transform that involves a 2D rotation, on the face image.

In order to perform the affine transform, the first step is to determine the center of face and the angle of rotation to be introduced. The angle of rotation $\theta_{rot}$ is the angle made by the line passing through the two eyes with the horizontal axis of the image frame. The face center is roughly estimated to be the point of intersection of the perpendicular from the mouth center to the line passing through the two eyes.

These corrections are done for every face image that has been trained into the model library, and for every query face image for which recognition is performed. The initial estimated locations of the eye and mouth feature points can be accordingly corrected around the face center to reflect the change in face orientation.
3.2.2 Find the best-fit locations of fiducial points

Active Shape Models (ASMs) are flexible shape models that can detect the periphery of objects belonging to a particular class for which they are trained, provided that the variation between the objects in that class is not very large. ASMs can actively change the shape of their contours to fit the object in an image, for which they are trained. An ASM can compensate for small variations in the object structure, as long as it has been trained for such variation. Variation in face pose causes significant variation in the distribution of the facial features. Hence, it is not possible to capture such variation with the help of a single ASM. If the training is insufficient, the ASM will almost certainly fail to converge to the correct face shape. Hence, to avoid facing the situation of having, this thesis work makes use of four distinct ASMs to cover a face pose variation between $0^\circ$ and $45^\circ$. 

![Figure 13. Face Alignment Process.](image)
Figure 14. The Different ASM Shapes Used For Covering Pose Angle Variation.

From the face pose parameter, which is a priori information, the appropriate ASM model is selected. Given the estimated feature points on the image after face alignment, it is possible to initialize the selected ASM model such that the scale, orientation and position of the model corresponds to the scale, orientation and position of an ASM to which the given estimated features belong. This process is exactly similar to the process used in model alignment while constructing the ASM model library in Section 2.1.2.

Once initialized, the ASM is made to run until it converges to the face shape in the image frame, or until an iteration count is reached. The next phase is to extract the node points in the ASM contour that will help align a face graph.

The ASM cannot define all points used for recognition. However, it can help place a face graph having all the required recognition points onto the image frame. A face graph is a geometric structure of nodes and connecting edges used in Elastic Bunch Graph Matching (see Chapter 2). An Elastic Bunch Graph collectively represents all the features from the set of trained faces which have the same pose angle as the query face image from which the feature points were extracted. The node location refinement process used in Elastic Bunch Graph Matching (EBGM) is an additional step taken to
ensure the proper placement of the feature points, prior to extracting the feature information from the face image frame.

Figure 15. Example of fitting the ASM to extract the required graph node positions: (a) Fitting ASM on face image, (b) Selecting points from ASM that are needed, and (c) Using extracted points to place face graph.

The number of nodes that are selected from the face graphs for feature extraction is common to all the face poses (Figure 13). These are twelve in number, and are listed in Table 1. It is important to note that even though the structure of the face graph connecting these feature points is the same for all face poses, the features themselves vary across the poses. Figure 13(c) indicates the extra nodes introduced by the face graph that are not available from the ASM. Hence, it is necessary to represent the collective set of faces with distinct face graphs as well. It is convenient to use as many face graph as there are ASM shapes (Figure 14).
Table 1. Description of the Feature Points used in recognition.

<table>
<thead>
<tr>
<th>Node Index</th>
<th>Label</th>
<th>Description;</th>
<th>Geometric Position on Face</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R.Eb</td>
<td>Right Eyebrow; Right Corner of Left Eyebrow</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>L.Eb</td>
<td>Left Eyebrow; Left Corner of Left Eyebrow</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>F.C</td>
<td>Face Center; Geometric Center between both Eyes</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>E.C</td>
<td>Eye Center; Center of Iris of Left Eye</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>N.C</td>
<td>Nose Center; Center of Nose</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>N.Tip</td>
<td>Nostril Tip; Left Tip of Left Nostril</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Cheek</td>
<td>Cheek; Center of Gravity of Left Cheek</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>U.Lip</td>
<td>Upper Lip; Upper Tip of Upper Lip</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>L.Corner</td>
<td>Lip Corner; Left Corner of the Mouth</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>L.Lip</td>
<td>Lower Lip; Lower Tip of Lowe Lip</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>L.E.Corner</td>
<td>Left Eye Corner; Left Corner of the Left Eye</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>R.E.Corner</td>
<td>Right Eye Corner; Right Corner of the Left Eye</td>
<td></td>
</tr>
</tbody>
</table>

Figure 16 Face Graphs Used for Different Face Poses

After the face graph is placed over the extracted features, the next last step is to refine the positions of the nodes so that the best-fit positions are obtained for extracting the feature descriptors. Heuristics are applied for some of the fiducial points that are easily describable, such as corners of eyes, eyebrows and lips. For more complex
features, such as upper lip and nostril, jet displacement estimation is used to provide the best-guess positions. It is important to note that the initial node position extracted from the ASM model might not always be within a six pixel error limit of the preferred node position. When this initial position used in jet displacement calculation has an estimation error greater than a six pixels, the resultant displacement calculation will be erroneous. Hence, jet displacement estimation is used only for some nodes that have a good initial placement via the ASM models.

![Figure 17. Example of finding best fit points around the eyes; (a) Initial Estimates, (b) Finding closest edges passing threshold, (c) Corrected Feature Points.](image)

**Figure 17.** Example of finding best fit points around the eyes; (a) Initial Estimates, (b) Finding closest edges passing threshold, (c) Corrected Feature Points.

![Figure 18. Example of finding Lip Corner By Finding the Steepest Edge in the neighborhood; (a) Initial Estimate, (b) Finding corner, (c) Corrected Feature Point.](image)

**Figure 18.** Example of finding Lip Corner By Finding the Steepest Edge in the neighborhood; (a) Initial Estimate, (b) Finding corner, (c) Corrected Feature Point.

Despite the fact that eye positions are given as *a priori* information, the accuracy of this information might vary depending on the procedure used to acquire them. Hence, best-fit estimations are done for the eye features using a heuristic search, described as follows.
The eye is typically a high-contrast region, with the contours of the eye being very distinctive. An edge map of the eye region would clearly indicate the periphery of the eyes. Furthermore, the structure of the eye is such that it tapers towards the side extremities. Hence, searching for the left and right extremities of the edge-map of the eye would directly provide the best-fit locations for the corner points of the eyes, as marked in Figure 17.

The lip corner is another feature point that can be located in a similar manner, since it is a high-contrast point and has a very low grayscale codevalue in its image neighborhood. Using the initial estimates from the ASM model as a proximity to the possible best-fit region, a search can be performed to find the high-contrast edge-map in the locality. The search finds the left tip of the edge-map, and confirms that it is the darkest point in its locality. This point is assumed to be the best-fit point for the lip corner, as indicated in Figure 18.

The heuristic searches explained above have been quite accurate in marking the feature points, when tested over a broad range of faces, and have thus been accepted as a usable heuristics for the mentioned feature points. The next section evaluates the performance of the automatic feature point placement scheme that has been introduced in this chapter.

3.3. Evaluation of the procedure

Direct numerical evaluation is not done for the automation of feature point selection. However, the performance of the algorithms is tested, and the comparison is performed for the automated feature selection scheme against the same test runs for manually selected feature points.
It is expected that the performance of the automatic feature selection scheme introduced in this chapter will perform worse than manual feature selection. However, the degradation is not expected to be large enough to negate the improvement in recognition by the steps introduced in this thesis work, as compared to the initial recognition procedures.

![Example of Automatic Feature Point Positioning in Comparison with Manually Selected Nodes.](image)

**Figure 19.** Example of Automatic Feature Point Positioning in Comparison with Manually Selected Nodes.

In order to be able to estimate the effects of automatic feature point placement for different recognition schemes, the evaluation of all the schemes is performed for both automatic as well as manual placement procedures. The graph in Figure 20 illustrates the effect of automatic feature point placement compared to manual feature point placement for the Direct Recognition scheme. Similar evaluations will be done for all the other schemes.

The performance degradation for the automatic feature point placement scheme introduced in this chapter seems to be acceptable. It can be noted that the automatic feature point placement works nearly as well as the manual placement for small face pose
angles. This is because the features are better defined in these smaller poses. As the face pose increases, the distortion in the features become much more prominent in comparison to the frontal faces as well as other faces in the same pose.

Figure 20. Recognition rates for Direct Recognition scheme.

This chapter has described a proposed method for automatically positioning the facial feature points, which is used in automating the face recognition schemes described in the upcoming chapters. Chapter 4 introduces some of the preliminary methods proposed by this thesis. Chapter 5 introduces a proposed recognition scheme that shows promising results for large face pose variation, and also modifies it using some of the methods introduced in Chapter 4.
Chapter 4  Feature Weighting and Face Derotation

The previous chapter introduced a procedure for automatically placing feature points required for face recognition. This procedure is utilized in the two different recognition schemes introduced in this chapter, each of which takes on a different approach to improving the recognition rates with variation in face pose. Section 4.1 reviews the basic recognition scheme that has been used in Elastic Bunch Graph Matching, which has been briefly covered in Chapter 2 as background information. This is pertinent to the recognition schemes proposed in this chapter. Section 4.2 introduces a scheme that makes up for some of the shortcomings of the basic recognition scheme. Section 4.3 introduces a completely different approach to trying to improve the face recognition across face pose.

Gabor wavelets are a common tool used in recognition schemes. The convolution of Gabor wavelets of different scales and orientations over a point in an image results in a set of complex coefficients which is called a ‘jet.’ A ‘jet’ is a representation of information contained in the neighborhood of the image point where the jet was constructed. This thesis uses a set of Gabor wavelets spanning five frequency scales and eight orientations, totaling to forty individual convolution results. Hence, any one jet will contain forty complex coefficients.

The coefficients in a jet vary with changes in the neighborhood information. For the same fiducial point in two different faces, the neighborhood information will be different. This creates a difference in the coefficients of the jets for the two images. This difference in value generally grows larger with larger visual difference. This concept is used in distinguishing features, and hence faces, from one another.
4.1. **Face Recognition Using Elastic Bunch Graph Matching**

Elastic Bunch Graph Matching represents faces as face graphs, with nodes linked with jets used as feature descriptors. The similarity function $S_a$, introduced in Section 2.2 of Chapter 2, for comparing two jets corresponding to points only a few pixels apart, tries to compare the closeness of each pair of coefficients from two jets $J$ and $J'$, and computes a similarity value. This similarity value of two jets can be as low as zero, if there is no correlation at all between the coefficients of the jets. Equation (2.12) representing the similarity function is repeated here for convenience:

$$S_a(J, J') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j^2 \sum_j a'_j^2}} \quad (4.1)$$

where $a_j, a'_j$ are the corresponding coefficients of jets $J$ and $J'$, and $j = 1, \ldots, 39$ is the coefficient index in the jets. Equation (4.1) can be used to compare node-wise similarity between any two nodes in any two graphs. When comparing feature similarity, Equation (4.1) is used for finding the similarity between corresponding feature nodes of any two graphs.

When comparing face similarity, Equation (4.1) is recursively used for each individual feature in the graphs of the face pair being compared. Recognition over an entire graph is performed as a similarity computation between jets per node, averaging the similarity value over the total number of nodes in the graph, and then selecting the
match that has the highest similarity score [9]. The equation for computing the similarity of two graphs, each having \( N \) corresponding nodes, is

\[
S_G(G^I, G^M) = \frac{1}{N} \sum_n S_a(G^{n,I}, G^{n,M}),
\]

where each node contributes an equal amount towards the whole face similarity value.

If there are \( K \) face graphs trained in the face library and a query is made, then an iterative comparison is performed between each of the \( K \) face graphs in the library and the face graph extracted from the query face image frame. The library face graph that has the highest similarity match value, against the query image face graph, corresponds to the best match. This is the basic recognition scheme used in Elastic Bunch Graph Matching [9]. This scheme is labeled as ‘Direct’ Recognition scheme in the comparison tests.

**4.2. Feature Weighting Scheme**

Section 4.1 reviewed the basic recognition scheme used in Elastic Bunch Graph Matching. As noted in the review, there is no discrimination on the facial features based on their ability to differentiate between faces. This section introduces a Feature Weighted Recognition Scheme, which studies the discrimination capability of the feature points used in face recognition and accordingly assigns a weight to each feature that affects the individual contribution of each feature towards the whole face recognition result.

**4.2.1 Motivation**

The face recognition scheme introduced in Elastic Bunch Graph Matching considers all nodes with equal weighting, thus giving no bias towards any facial features. In general,
some features on the face are more distinguishing than others. For frontal face recognition, it is understandable that features such as the eyebrows, or eyes and mouth are much more distinguishing than other regions of the face. However, for recognition of faces across pose variation, another factor that plays an important role in considering the distinguishing features is the amount of deformation the features undergo over pose variation. Features that vary dramatically with variation in pose might not be suitable for identifying/distinguishing individuals in a group. Hence, a possible consideration is to give more importance to features that can discriminate faces properly at a particular face pose. Effectively, a different set of weights can be used while calculating the total face similarity for different face poses.

4.2.2 Procedure

As previously mentioned, the Direct Recognition scheme selects the face match that has the highest similarity value during comparison. This scheme does not consider the individual contributions of each node in the graph. The following figures show the individual node similarity values of a correct face match at different face poses for one individual in the library.

It can be inferred from Figure 22 that different graph nodes do not contribute equally to face recognition. Furthermore, the similarity value of each feature node varies with variation in face pose. Some nodes have comparable similarity values for correct and incorrect matches. Such nodes would not contribute much towards the final recognition rates.
Figure 22. Face Similarity of a sample face query image against the frontal face model in the library, with query face image at 15°, 30°, and 45° poses.

However, some nodes have very distinct difference in similarity between the correct matches and incorrect matches. Giving more importance to these nodes might result in better recognition rates. Hence, it might be advantageous to numerically weigh the nodes according to their importance.

For convenience of representation, let $S_{i,j}^n$ be the notational equivalent for the similarity of the jets corresponding to the $n^{th}$ node of two face graphs, labeled $i$ and $j$.

$$S_{i,j}^n = S_a(J^{n,i}, J^{n,j})$$ (4.2)
where $S_a(J^{n,i},J^{n,j})$ is the similarity function between two given jets, defined in Equation (4.1).

In the notation used in Equation (4.2), it can be interpreted that if the face graph labels $i$ and $j$ point to one individual’s library and query face graphs, then Equation (4.2) is simply computing the similarity match value of the ‘correct match’. In a similar manner, if the labels $i$ and $j$ do not point to different face graphs of the same individual, the computed value can be considered as a similarity match value of an ‘incorrect match’.

The notation $S_{i,j}^{n}$ used in Equation (4.2) is for similarity between two jets or features. Let this notation be extended to entire face graphs by following the notation $S_{i,j}$, where an absence of index $n$ indicates that whole graph is represented.

Comparing a query face with each face in a library set of $K$ faces will result in one correct match and $K-1$ incorrect matches. Each of these matches will have a face graph similarity value $S_{k,Q}$, where the label $k$ points to the $k^{th}$ face in the library set, and label $Q$ point to the query face graph. Preferably, the face corresponding to the highest value of $S_{k,Q}$ would be the correct match. However, the highest similarity match value may not always correspond to the correct match. It would be possible to get a more preferable result, if the nodes making a larger contribution to the difference in similarity with incorrect matches are given a larger weighting.

Given a set of training face images, for which the correct matches are known, it is possible to simulate correct matches and incorrect matches and obtain information on the contribution of each feature node. The training set will contain $K$ faces of a given pose, say $15^\circ$, and the corresponding correct matches for each of the $K$ faces at frontal face pose. In this condition, the correct match $S_{k,Q}$ corresponds to the similarity match when
both the face indices are the same and an incorrect match is for any other index pair. Then the variance

\[ \sigma_{n,\text{correct}}^2 = \sigma^2 \left( \{ s_{n,q}^n : q = 1, \ldots, K \} \right) \]  

(4.3)

represent the variance in similarity of the \( n \)th node for a correct match, and

\[ \sigma_{n,\text{incorrect}}^2 = \sigma^2 \left( \{ s_{p,q}^n : p \neq q; p, q \in (1, \ldots, K) \} \right) \]  

(4.4)

is the variance in similarity of the \( n \)th node for incorrect matches. be vector

The variance of correctly matched nodes indicates how reliable the nodes are in distinguishing features correctly, and the variance in incorrectly matched nodes indicates how unreliable the nodes are in discarding wrong matches. Hence a suitable set of weights would take the two variances into account. Fischer’s criterion can be applied in this context to calculate the weights using the two variances. Then, the weight for the \( n \)th node would be

\[ W_n = \frac{\sigma_{n,\text{correct}}^2}{\sigma_{n,\text{incorrect}}^2}. \]  

(4.5)

The following graph plots the weights used in this work, for each of the nodes for the face graph structure implemented in face recognition. The weights \( \{ W_n \} \) have been normalized to get \( \{ W'_n \} \) such that they total to a value of 1. This is just to avoid scaling up the similarity value.

\[ W'_n = \left( \frac{W_n - \min_n(W_n)}{\max_n(W_n) - \min_n(W_n)} \right) \left( \frac{9}{10} \right) + 0.1 \]  

(4.6)

Equation (4.6) ascertains that every feature node contributes to at least 10% of their similarity value, regardless of the calculated weights. Table 2 tabulates the effective weights that have been calculated using Equations (4.3), (4.4), (4.5) and (4.6) for each
feature point in the face graph used for the face recognition step. Figure 24 graphically represents the weights in Table 2.

![Normalized Weights]

**Figure 24.** Feature weights calculated for different pose graphs.

In Figure 24 and Table 2, the weights have been scaled down such that the sum of the weights of all features is equal to a value of 1.0, so as to avoid having the similarity value of the whole face multiplied by a scaling factor.

It can be noted that different features carry varying weights as the face pose changes. It is further notable from the table that certain features, such as the eye center, have a more distinguishable characteristic than others.
Table 2. Calculation of weights for 15°, 30° and 45° Face Pose Graph for a Sample Run using Equation (4.6).

<table>
<thead>
<tr>
<th>Node # (n)</th>
<th>Label</th>
<th>Calculated Weights, $W_n^N$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>15° Pose</td>
</tr>
<tr>
<td>1</td>
<td>R.Eb</td>
<td>0.098</td>
</tr>
<tr>
<td>2</td>
<td>L.Eb</td>
<td>0.023</td>
</tr>
<tr>
<td>3</td>
<td>F.C</td>
<td>0.075</td>
</tr>
<tr>
<td>4</td>
<td>E.C</td>
<td>0.102</td>
</tr>
<tr>
<td>5</td>
<td>N.C</td>
<td>0.045</td>
</tr>
<tr>
<td>6</td>
<td>N.Tip</td>
<td>0.055</td>
</tr>
<tr>
<td>7</td>
<td>Cheek</td>
<td>0.031</td>
</tr>
<tr>
<td>8</td>
<td>U.Lip</td>
<td>0.107</td>
</tr>
<tr>
<td>9</td>
<td>L.Corner</td>
<td>0.115</td>
</tr>
<tr>
<td>10</td>
<td>L.Lip</td>
<td>0.099</td>
</tr>
<tr>
<td>11</td>
<td>L.E.Corner</td>
<td>0.125</td>
</tr>
<tr>
<td>12</td>
<td>R.E.Corner</td>
<td>0.124</td>
</tr>
</tbody>
</table>

From Figure 24, it can be deduced that some features have relatively lower generated weights at certain face pose angles than at other face pose angles. The lower lip (L.Lip) feature descriptor loses its significance for 45° face pose. This is primarily because the feature point lies along the face contour at a 45° face pose, while it is significantly distant from the periphery at smaller pose angles. This creates a tremendous difference in neighborhood information between the larger and smaller face poses, and hence almost incapacitates the feature from being able to perform recognition. As a result, lower weighting is given to that feature.

It can also be noted from Figure 24 that some features do not have as much weight in lower face poses than they do in larger face poses, where features typically become more distorted. This can be justified by understanding that the other features have lost their ability to discriminate very well, because of distortion due to pose variation. In other words, it can also be considered that some features did not get
distorted as much as the other features, and hence have higher generated weights than those other features.

To implement the weights calculated in this section, the equation for computing the similarity of two graphs, each having $N$ corresponding nodes, has to be modified as:

$$S_g(G^I, G^M) = \frac{1}{N} \sum_{n=1}^{N} W'_n \cdot S_a(J^{n,I}, J^{n,M})$$

(4.7)

It is to be noted that training is required to generate the weights suitable for a set of faces, as the weights might be different for different classes of faces. Also, the weights might vary with respect to the pose variation, as different facial features will not have the same similarity contribution as the face pose varies. Hence, the generation of these weights will require training prior to use in the recognition scheme.

This subsection described the procedure for implementing a proposed weighted recognition scheme, where features with better discriminating ability than others have been given higher weighting. The next subsection looks into evaluating the results obtained from the proposed scheme and makes interpretation on the utility of the proposed scheme.

### 4.2.3 Evaluation of Weighted Recognition Scheme

The evaluation of the proposed recognition scheme is carried out by examining the average improvement in face recognition hit rate. Face recognition is performed, independently for each face pose, on a database of 100 face pairs, a portion of which will be exclusively used for training. Each face pair contains one image of an individual’s face in frontal pose, which is stored in the library, and another image in the required non-frontal pose.
Since training is involved, it may not be ideal to use only a selective set of faces, as it might not always be the representative of a test set. Hence, a k-fold cross-validation approach is adopted, with the database divided into five discrete subsets. The k-fold cross-validation uses one subset for testing, and the remaining for the required training. Thus, 80 face pairs are used for training, and 20 face pairs used for testing.

Training involves calculating the feature weights as described in the previous section. Testing is done by taking a non-frontal face from the test set and running a similarity match against all the frontal faces in the test set to find a match. If the match is a correct match, it is considered as a hit, and if it is an incorrect match, it is a miss. Recognition rate is calculated as the percentage of hits in the total test set.

The evaluation of the Weighted Recognition scheme is done by determining the change in the recognition hit rate for the Direct Recognition scheme as well as the Weighted Recognition scheme over the same test set of faces. This is done using both manual and automatic feature point placement, as explained in Chapter 3.

The results from the evaluation performed are tabulated in Tables 3, through 8. Tables 3-5 tabulate the recognition results from manual feature point placement, while Tables 6-8 tabulate the results from the scheme using automatic feature point placement process explained in Chapter 3. Interpretation of results follows the tables.
Table 3. Evaluation Results for Weighted Face Recognition scheme using Manually Positioned Feature Points for 15° Face Pose Query Images.

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Weighted # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>18 (90%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>19 (95%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>97%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 4. Evaluation Results for Weighted Face Recognition scheme using Manually Positioned Feature Points for 30° Face Pose Query Images.

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Weighted # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>10 (50%)</td>
<td>14 (70%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>16 (80%)</td>
<td>19 (95%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>18 (90%)</td>
<td>19 (95%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>13 (65%)</td>
<td>16 (80%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>18 (90%)</td>
<td>18 (90%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>75%</td>
<td>86%</td>
</tr>
</tbody>
</table>

Table 5. Evaluation Results for Weighted Face Recognition scheme using Manually Positioned Feature Points for 45° Face Pose Query Images.

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Weighted # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>7 (35%)</td>
<td>8 (40%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>12 (60%)</td>
<td>14 (70%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>11 (55%)</td>
<td>11 (55%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>11 (55%)</td>
<td>13 (65%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>15 (75%)</td>
<td>16 (80%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>56%</td>
<td>62%</td>
</tr>
</tbody>
</table>
Table 6. Evaluation Results for Weighted Face Recognition scheme using Automatic Feature Points Positioning for 15° Face Pose Query Images.

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Direct # (%)</th>
<th>Weighted # (%)</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td></td>
<td>17 (85%)</td>
<td>18 (90%)</td>
<td>5%</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td></td>
<td>20 (100%)</td>
<td>20 (100%)</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td></td>
<td>20 (100%)</td>
<td>20 (100%)</td>
<td>0%</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td></td>
<td>19 (95%)</td>
<td>20 (100%)</td>
<td>5%</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td></td>
<td>17 (85%)</td>
<td>18 (90%)</td>
<td>5%</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td></td>
<td>93%</td>
<td>96%</td>
<td>3%</td>
</tr>
</tbody>
</table>

Table 7. Evaluation Results for Weighted Face Recognition scheme using Automatic Feature Points Positioning for 30° Face Pose Query Images.

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Direct # (%)</th>
<th>Weighted # (%)</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td></td>
<td>9 (45%)</td>
<td>11 (55%)</td>
<td>10%</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td></td>
<td>12 (60%)</td>
<td>14 (70%)</td>
<td>10%</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td></td>
<td>14 (70%)</td>
<td>17 (85%)</td>
<td>15%</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td></td>
<td>14 (70%)</td>
<td>16 (80%)</td>
<td>10%</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td></td>
<td>13 (65%)</td>
<td>14 (70%)</td>
<td>5%</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td></td>
<td>62%</td>
<td>72%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Table 8. Evaluation Results for Weighted Face Recognition scheme using Automatic Feature Points Positioning for 45° Face Pose Query Images.

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Direct # (%)</th>
<th>Weighted # (%)</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td></td>
<td>7 (35%)</td>
<td>8 (40%)</td>
<td>5%</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td></td>
<td>9 (45%)</td>
<td>10 (50%)</td>
<td>5%</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td></td>
<td>10 (50%)</td>
<td>12 (60%)</td>
<td>10%</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td></td>
<td>11 (55%)</td>
<td>12 (60%)</td>
<td>5%</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td></td>
<td>9 (45%)</td>
<td>11 (55%)</td>
<td>10%</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td></td>
<td>46%</td>
<td>53%</td>
<td>7%</td>
</tr>
</tbody>
</table>
By direct comparison of the results in the above tables, it can be seen that the Weighted Recognition scheme has better recognition hit rate than the Direct Recognition scheme, which implies that the proposed recognition scheme in this section is capable of improving recognition over variation in pose. The following graph illustrates the information in the tables graphically.

![Recognition Rates of Implemented Algorithms](image)

**Figure 25.** Recognition Rates for Weighted Recognition scheme.

It can be further noted from Figure 25 that the improvement in recognition for the manual version of the recognition schemes is larger for smaller pose angles, compared to larger angles, e.g. 30° compared to 45°. This can be explained by noticing that features are more deformed in a 45° face pose compared to the deformation in a 30° face pose, which directly affects how much the features would be able to contribute towards recognition.

This section introduced a recognition scheme where existing feature descriptors are weighted according to how capable they are at discriminating faces. This scheme does not look into improving the available feature descriptors to reduce the effect of face pose...
variation, but rather relies on features that haven’t been affected so much by the face pose variation. Although this is reasonable, it is not sufficient to improve the recognition rates. A more active approach to improving the recognition rates would be to directly impact change in the feature descriptors to counter the effect of face pose variation. The next section suggests a method for improving the recognition rates by means of changing the feature descriptors themselves prior to recognition.

4.3. Face Derotation Scheme

The previous section examined improving recognition rates by considering each face pose individually, and giving larger weights to those features that could better distinguish between the correct and incorrect faces. This section takes a more active approach, by trying to perform face derotation to transform the features back to how they were prior to the face pose variation. The following sub-sections describe the motivation involved and the proposed procedure for doing so. The last section performs an evaluation of the proposed scheme in this section.

4.3.1 Motivation

The human face is a complex three-dimensional (3D) structure, and representing the face on a two-dimensional (2D) image frame, with uniformity across all the captured face images, requires that some constraints be applied on the relative positions of the face and the image plane. To an extent, it is possible to control the pose variation of the face with respect to the image frame. Complex 3D structures can rotate along three axes, and this can be described by Euler angles. However, for the purpose of this work, the human face is to be restricted to rotation around the vertical axis. As with the rest of the work in this
thesis, this restriction in movement gives control over face pose angle variation while keeping the other variables constant, giving the opportunity to counter only the variation under concern.

Figure 26. An illustration for the concept of Derotation.

The face pose variation can be viewed as if a picture of a human face is pasted on the curved side of a cylinder and the cylinder is rotated around its axis, creating a variation in face pose angle as the face turns away from the viewing plane. Theoretically, it is possible to skew a 2D image on the cylinder’s surface using simple 2D image transforms to recover a frontal-looking view of the 2D image, without significant distortion from the pose angle variation. Modeling the human face appearance to be a flat 2D image pasted on the side of a cylinder, it might be possible to construct a similar
image transformation to synthesize a frontal view of a face at another given pose. However, due to the 3D nature of the face, the synthesis only works for small pose angles.

### 4.3.2 Procedure

The process of synthesizing a frontal image from a pose view of a face, or ‘derotating’ the face from a pose view to its frontal view is labeled as Derotation.

![Derotation Illustration](image)

**Figure 27.** Illustration for the Concept of Derotation.

The derotation process can be considered as one in which the points along the face center are transformed to fall into a straight vertical line, while the rest of the face features are transformed along with the face profile. The ‘derotated’ face image is expected to resemble the original frontal view image of the face, which could give a much better recognition hit rate when compared to the frontal face library images.
There are two steps to performing face derotation. The following sub-section details the first of the two steps, which involves the setting up of control points, and the latter sub-section explains the derotation process which makes use of these control points.

### 4.3.2.1 Control Points

The process of derotating a face image relies on aligning, or mapping, the current points that constitute the face profile into a straight vertical line that will form the face profile of the synthesized frontal face image. This mapping of image points from the initial positions onto destination positions, such that the entire face image aligns, is the same type of mapping done in an image registration process. Image registration is the process of ‘linking’ two or more images, by trying to align known features, called control points, which are common in the images being registered. In this specific application, the control points include the set of points along the current face profile of the non-frontal pose face, which are mapped onto corresponding points on an estimated synthesized face profile.

The source control points are obtained by the automatic feature point placement procedure (explained in Chapter 3) on the query image frame, which has the non-frontal pose view of the face to be recognized. However, the graph structure used for obtaining the source control points is slightly different compared to the graph used for face recognition, and covers more area on the face, to be able to obtain a good set of control points.
Without sufficient training, the automatic feature point positioning does not give good placement results when there is high frequency information in the image. Hence most of the face graph nodes on the farther side of the face profile are ignored while selecting a good set of control points.

The set of source control points is marked in Figure 26. However, to be able to synthesize the derotated face, it is necessary to estimate the target control points as well. For this purpose, several frontal face graphs have been used to estimate a generic frontal face graph structure that would fit all the face graphs in the set equally well. However, since every face is different from the others, some flexibility has to be accounted for.
while mapping the control points to the generic structure. A generic face graph specific to the pose is used as a reference to determine the approximate variation in some general parameters such as width-to-height ratio of the face graph. These parameters are then applied to the generic frontal face graph to get a better approximation of the control point distribution in the face to be synthesized.

The control points are used for generating a second-order polynomial image transformation that maps the control points from the source image onto the target control points of the synthesized image frame. The next section describes the proposed image transformation process.

### 4.3.2.2 Image Transformation

When the relationship between the source and target control points is represented by a set of polynomial equations, then the transformation used for mapping the control points is a polynomial transformation. If the source control points are represented by the coordinate set \((u, v)\) which is mapped onto a target coordinate system \((x, y)\), the second-order polynomial relationship is given by the following equations.

\[
\begin{align*}
    u &= C_{11} + C_{12}x + C_{13}y + C_{14}xy + C_{15}x^2 + C_{16}y^2 \\
    v &= C_{21} + C_{22}x + C_{23}y + C_{42}xy + C_{25}x^2 + C_{26}y^2
\end{align*}
\]

where \(C_{mn}\) are unknown constants. To be able to define the mapping, each of the twelve constants need to be solved. At least six control-point pairs are required to be able to solve for the coefficients in the above mapping.

The polynomial image transform required is generated with the help of functions available in the MATLAB Image Processing toolkit for Image Registration. The
following MATLAB code segment shows the library function calls made to be able to
generate the image transform, and synthesize the frontal face image.

```matlab
% CP2TFORM calculates the transformation required to map the
% Source_Control_Points to the Target_Control_Points using
% the preferred method, which, in this case, is the second-order
% polynomial transformation
T = cp2tform( src_ctrl_pts, target_ctrl_pts, 'polynomial', '2');

% The transformation calculated can then be directly applied
% as an image transformation, using the IMTRANSFORM library call.
derotated_Face = imtransform( query_Face, T );
```

**Figure 30** Derotation Process: (a),(d): Initial Query Face at 22° pose; (b),(e): Polynomial Transformation using mapped control points; (c),(f): Final Image after mirroring at face profile.
The amount of information available in the face image after the derotation process cannot be more than what is available prior to the process. Hence image data pertaining to the face on the farther side of the profile cannot be improved to reconstruct the exact original image. However, the human face is generally symmetric along the vertical axis, and can be mirrored along the axis with very little change.

![Comparison of the derotated face to the preferred face in library](image)

**Figure 31** Comparison of the derotated face to the preferred face in library

Once the query face image has been derotated, the feature points have to be selected again on the synthesized image. Since the target control points are known, it is possible to estimate the rough locations of the feature points in the synthesized image. Finally, using the automatic feature point positioning scheme described in Chapter 3, the feature points for the new image are estimated, and the feature descriptors are extracted. It is expected that the features extracted after face derotation would result in better matches in comparison to the features extracted directly from the non-frontal face poses. Face recognition is performed using Equation (2.16) described in Section 2.3.2, since both the face graphs being compared are frontal face graphs.
4.3.3 Evaluation of Derotation Results

The previous sub-sections detailed a process proposed for obtaining higher recognition rates by deforming the face image to imitate a Face Derotation process. This section performs an evaluation of the proposed scheme, and derives conclusions based on the results obtained.

The evaluation of the proposed recognition scheme is carried out by examining the average improvement in face recognition hit rate. The face recognition is performed independently for each face pose on a database of 80 face pairs, a portion of which will be exclusively used for training. Each face pair contains one image of an individual’s face in frontal pose, which is stored in the library, and another image in the required non-frontal pose.

In order to conform to the evaluation of other schemes in this work, the k-fold cross-validation approach is adopted, with the database divided into subsets of 20 face pairs. Hence, 20 face pairs are used for testing at any time. Testing is done by taking a non-frontal face from the test set and running a similarity match against all frontal faces in the test set to find a match. If the match is a correct match, it is considered as a hit, and if it is an incorrect match, it is a miss. Recognition rate is calculated as the percentage of hits in the total test set.

The evaluation of recognition based on the Face Derotation scheme, labeled in the results as Derotated Recognition, has been carried out for $22^\circ$ poses only. The recognition tests have been carried out in the same fashion as the tests for other recognition schemes in this work, for both manual and automatic feature placement.
Table 9 Evaluation Results for Derotated Face Recognition scheme using Manual Feature Points Positioning for 22° Face Pose Query Images

| Run Number | # of Test Faces | Face Recognition | | | |
|---|---|---|---|---|
| | | Direct # (%) | Derotated # (%) | Improvement % |
| 1 | 20 | 17 (85%) | 18 (90%) | 5% |
| 2 | 20 | 19 (95%) | 19 (95%) | 0% |
| 3 | 20 | 18 (90%) | 18 (90%) | 0% |
| 4 | 20 | 19 (95%) | 20 (100%) | 5% |
| AVERAGE | | 91.25% | 93.75% | 2.5% |

Table 10 Evaluation Results for Derotated Face Recognition scheme using Automatic Feature Points Positioning for 22° Face Pose Query Images

| Run Number | # of Test Faces | Face Recognition | | | |
|---|---|---|---|---|
| | | Direct # (%) | Derotated # (%) | Improvement % |
| 1 | 20 | 16 (80%) | 16 (80%) | 0% |
| 2 | 20 | 18 (90%) | 17 (85%) | -5% |
| 3 | 20 | 17 (85%) | 18 (90%) | 5% |
| 4 | 20 | 19 (95%) | 18 (90%) | -5% |
| AVERAGE | | 87.5% | 86.25% | -1.25% |

From Tables 9 and 10, it is notable that this recognition scheme does not seem to provide any improvement over the Direct Recognition scheme. Hence, justifying the results is important to understanding the applicability of this scheme. Further analysis of the results showed the following:

(a) Query faces that had other elements of rotation present and did not conform to the face poses of the rest of the faces in the test set, generated poorly derotated face image, which degraded the recognition of the query faces, but did not improve results. The face database used for the testing was not consistent enough.
(b) The transformation performed on the target face graph, which is used for mapping the target control points, is not flexible enough to adapt properly to the set of training face graphs. As a result, the structure of the target face graph might not often resemble a preferred face structure for the test set faces. This is because of the requirement to map onto a generic frontal face graph. Faces can be very different in structure, and the frontal face graph map needs to reflect these differences effectively.

The above two issues account for the poor overall recognition obtained. However, it might be possible to observe better results, if the rigidity of the current image deformation scheme was reduced by making use of the training set of faces to construct a generic target face grid that is relevant to the face structure in the test set.
Figure 32 Derotation Examples
Chapter 5  Pose-invariant Face Recognition with Jet-Mapping

The previous chapter detailed some of the preliminary steps that were used for improving face recognition results across pose. Although the Feature Weighting scheme provided some improvement in recognition rates, it was insufficient. The Face Derotation scheme, introduced in the previous chapter, proved to be much less useful in its current state. This chapter introduces the implementation of a newly proposed mapping scheme focused on improving recognition of faces across poses by altering the feature descriptors, rather than altering the features themselves as was done in the Face Derotation scheme. Although the concept implemented here can work for comparison between faces for any known pose variation, the focus in this work is placed on trying to improve recognition rates of non-frontal faces against their frontal face matches. The following sections explain and evaluate this recognition scheme.

Section 5.1 explains the motivation behind the proposed scheme, while Section 5.2 details the proposed procedure. Section 5.3 performs an evaluation of the scheme proposed in Section 5.2. Section 5.4 extends the scheme proposed in this chapter by combing it with other schemes proposed earlier. Section 5.5, then, performs a final evaluation of the extended scheme proposed in Section 5.4.

5.1. Motivation

Controlled face pose variation may be assumed to create a controlled deformation in face image information. If this assumption is true, then it should be possible to estimate this deformation, and find a mapping that helps compensate for it. The facial feature
descriptors used in this work are the Gabor-wavelet based ‘jets’ that generate an array of complex coefficients as the result of convolving the image region with a series of Gabor wavelets. Deformation in the face image due to pose would affect the convolution results stored as jets. Hence, it is possible to relate the variation in jet information to the deformation of face image due to pose.

5.2. Procedure

This section describes the proposed recognition scheme in two steps. The first step, explained in Section 5.2.1, is to understand the relation between the same feature descriptors across different face poses. This understanding is used in Section 5.2.2 to generate a mapping that helps transform the feature descriptors to provide better face recognition.

5.2.1 Study of Jet Coefficients

The general face recognition scheme here focuses on the similarity match between pairs of jets. While performing recognition of a non-frontal face in an image frame, a face-by-face similarity check is performed, where the jets extracted for the non-frontal face graph nodes are compared with their corresponding jets in the frontal face graph with the similarity function $S_a(J, J')$ from Equation (2.12). Higher similarity value for a correct match is likely to result in selecting the correct match being. Equation (2.12) is reproduced here for convenience.

$$S_a(J, J') = \frac{\sum_j a_j a_j'}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}$$  \hspace{1cm} (5.1)
In the above equation, $a_j$ and $a'_j$ are the magnitudes of $j^{th}$ complex coefficients of the respective jets $\mathcal{J}$ and $\mathcal{J}'$, when represented in polar coordinates. The function $S_a$ can be compared to finding the correlation between two vectors – the lesser the similarity, or correlation, between the two vectors, the lower is the similarity value.

As the objective of the recognition scheme is to find the set of face graphs that has the highest similarity with the query face graph, improving the correlation between the correct matches and the query face graph would possibly result in better recognition rates. For this, it becomes necessary to have a closer look at some training data for the face poses under consideration.

The study of the training data can be done separately for each face pose collection. Within the set of faces with a common pose, there are a common number of nodes, each node having a jet linked with it. In order to be able to study all this data, and be able make sense of the information available, the data has to be studied systematically.

It is sensible to consider the consistency of jets across faces pertaining to a particular pose. Again, each of the 40 coefficients has to be compared with its counterpart in the other jets. Figures 33(a) and 33(b) incorporate collective information for one complex coefficient of a given feature descriptor jet for a collection of face graphs at a given pose. In the graphs above, the circles represent the standard deviation of the distribution, centered at their means. These data distributions can be compared to study the effect of face pose variation.
5.2.2 Coefficient Mapping

This section describes the process involved in creating a mapping of the complex coefficients in the feature descriptor jets that can improve the correlation between the feature descriptors across face pose. The data distributions observed suggest a few properties that can be used to correlate the jets across pose variation.

Let $\mu_{n,j}$ be the mean magnitude of the distribution for the $j^{th}$ coefficient of the descriptor jet for the $n^{th}$ node in the face graph, and $\sigma_{n,j} = \sigma(a^n_j)$ be the standard deviation of the same. Furthermore, let $\mu_{n,j,POSE}$ and $\sigma_{n,j,POSE}$ be the mean and standard deviation at the given pose.

From having studied the data distribution, it is possible to suggest that shifting the distribution of the non-frontal face pose to better overlap with the frontal face pose distribution would help improve the correlation between the jets. Furthermore, as the
similarity function discards any phase information, it would be sufficient if only the magnitude of the coefficients were altered.

![Diagram of the recognition process]

**Figure 34** Jet-Mapped Recognition Scheme

Using a training set of suitable size, the means and standard deviations of the jet coefficient distributions for the query face pose and the target library face pose are calculated as \((\mu_{n,j,POSE}, \sigma_{n,j,POSE})\) and \((\mu_{n,j,Frontal}, \sigma_{n,j,Frontal})\) respectively. Then, the proposed mapping is applied on the jets \(\{J^n\}\) from the query face graph to obtain mapped jet coefficients. The calculation for this mapping is as follows:

\[
J_{j, \text{mapped}}^n = M_j^n(J_j^n) = \left( J_j^n - \mu_{n,j,POSE} \right) \cdot \left( \frac{\sigma_{n,j,Frontal}}{\sigma_{n,j,POSE}} \right) + \mu_{n,j,Frontal} \quad (5.2)
\]

and

\[
\mathcal{J}^n_{\text{mapped}} = \{J_{j, \text{mapped}}^n : j = 1 \ldots 40\}
\]

This mapping is performed on the feature descriptors prior to other steps in the recognition process. The following figures illustrate how this proposed mapping affects the distributions in the previously illustrated figures.

This suggested mapping is meant to improve the correlation between the query face graph feature descriptors and the face graph feature descriptors in the library. The face recognition function updated to reflect the above mapping is

\[
S_G(G^l, G^M) = \frac{1}{N} \sum_{n=1}^{N} S_a(J_{\text{mapped}}^n, J^n_M)
\]
5.3. **Evaluation of Jet Mapping Scheme**

The evaluation of the recognition scheme proposed in the previous section is carried out in this section by examining the average improvement in face recognition hit rate. The face recognition is performed, independently for each face pose, on a database of 100 face pairs. Each face pair contains one image of an individual’s face in frontal pose, which is stored in the library, and another image in the required non-frontal pose. A k-fold cross-validation approach is adopted, with the database divided into 5 discrete subsets. The k-fold cross-validation uses one subset for testing, and the remaining for the required training. Thus, 80 face pairs are used for training, and 20 face pairs used for testing. Training involves calculating the feature weights as described in the previous
section. Testing is done by taking a non-frontal face from the test set and running a similarity match against all the frontal faces in the test set to find a match.

The evaluation of the Weighted Recognition scheme is done by determining the change in the recognition hit rate for the Direct Recognition scheme as well as the Weighted Recognition scheme over the same test set of faces. This is done using manual feature point placement, and using automatic feature point placement explained in Chapter 3. The results are tabulated in Tables 11-13 and Tables 14-16.

**Table 11** Evaluation Results for Jet-Mapped Face Recognition scheme using Manually Positioned Feature Points for 15° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Jet-Mapped # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>19 (95%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>19 (95%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>98%</td>
<td>100%</td>
</tr>
</tbody>
</table>

**Table 12** Evaluation Results for Jet-Mapped Face Recognition scheme using Manually Positioned Feature Points for 30° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Jet-Mapped # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>10 (50%)</td>
<td>13 (65%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>16 (80%)</td>
<td>17 (85%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>18 (90%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>13 (65%)</td>
<td>16 (80%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>18 (90%)</td>
<td>18 (90%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>75%</td>
<td>84%</td>
</tr>
</tbody>
</table>
### Table 13 Evaluation Results for Jet-Mapped Face Recognition scheme using Manually Positioned Feature Points for 45° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Jet-Mapped # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>7 (35%)</td>
<td>9 (45%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>12 (60%)</td>
<td>16 (80%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>11 (55%)</td>
<td>17 (85%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>11 (55%)</td>
<td>15 (75%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>15 (75%)</td>
<td>15 (75%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>56%</td>
<td>72%</td>
</tr>
</tbody>
</table>

### Table 14 Evaluation Results for Jet-Mapped Face Recognition scheme using Automatic Feature Point Positioning for 15° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Jet-Mapped # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>17 (85%)</td>
<td>19 (95%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>19 (95%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>19 (95%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>19 (95%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>96%</td>
<td>97%</td>
</tr>
</tbody>
</table>

### Table 15 Evaluation Results for Jet-Mapped Face Recognition scheme using Automatic Feature Point Positioning for 30° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Jet-Mapped # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>11 (55%)</td>
<td>13 (65%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>13 (65%)</td>
<td>15 (75%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>13 (65%)</td>
<td>17 (85%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>12 (60%)</td>
<td>17 (85%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>13 (65%)</td>
<td>16 (80%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>62%</td>
<td>78%</td>
</tr>
</tbody>
</table>
Table 16 Evaluation Results for Jet-Mapped Face Recognition scheme using Automatic Feature Point Positioning for 45° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Jet-Mapped # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>8 (40%)</td>
<td>12 (60%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>7 (35%)</td>
<td>11 (55%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>9 (45%)</td>
<td>12 (60%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>11 (55%)</td>
<td>15 (75%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>11 (55%)</td>
<td>13 (65%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>46%</td>
<td>63%</td>
</tr>
</tbody>
</table>

From Tables 11-13, that tabulate the recognition results of the Jet-Mapping Recognition scheme using manually placed feature points, it can be noted that the recognition rate improves with increase in pose angle variation, as opposed to the Feature Weighting Recognition scheme, where performance decreases with increase in face pose angle variation. This can be justified by stating that the Jet-Mapping scheme improves recognition capability by modifying all feature descriptors towards better recognition contribution, while the Feature Weighting scheme only made use of the features that could help recognition in a more passive manner. This improvement is also noticeable in the results obtained for automatic feature point placement listed in Tables 14-16. The following graph further illustrates the results in the table graphically, which mark the dramatic improvement in recognition rates in both automatic and manual placement scenarios.
5.4. Combination of Jet Mapping and Weighting Schemes

It has been noted in the previous section that the Jet-Mapped Recognition scheme has good improvement in recognition rate, by having improved the correlation between the features from the query face image, and the features of the face graphs stored in the library. This section justifies and describes how another scheme, introduced in Chapter 4, can be implemented along with the recognition scheme introduced in the previous section, to provide greater improvement as a combined effort of the two schemes.

Chapter 4 introduced a Feature Weighting scheme that showed noticeable improvement in recognition rates. The Weighted Recognition scheme tried to focus on increasing the weight of the features that were able to discriminate the faces better than others. The Jet-Mapped recognition scheme, on the other hand, has been trying to
improve the similarity between the correct face matches. Since these two schemes place efforts in different concepts that do not overlap with each other, it is possible to combine both of these schemes in order to improve their recognition rates. The remaining of this section describes how this combination is carried out.

**Figure 37 Combined Recognition Scheme**

Let $T_i^n(\cdot)$ be the mapping function for the $j^{th}$ coefficient of the jet descriptor of the $n^{th}$ node in the query face graph, and $W_n$ be the weight proposed for the $n^{th}$ face feature descriptor. To determine the mapping function $T_i^n(\cdot)$ detailed in Section 5.2, a training face set is used for the poses concerned. After the coefficient mapping is evaluated and applied on the training face set, the same face set is used for computing a set of normalized weights $W'_n$ that will be used for improving recognition per Section 4.2.

The first step involves mapping the jets from the query image to a new set of jets

$$\mathcal{J}_{j,\text{mapped}}^n = T_j^n(\mathcal{J}_j^n); \mathcal{J}_{\text{mapped}}^n = \{\mathcal{J}_{j,\text{mapped}}^n : j = 1 \ldots 40\}$$

$\mathcal{J}_{\text{mapped}}^n$ is used in combination with weights for the actual face recognition stage using Equation (4.6). The modified graph similarity function used for recognition can be described as

$$S_G(G^I, G^M) = \frac{1}{N} \sum_{n=1}^{N} W'_n \cdot S_a(\mathcal{J}_{\text{mapped}}^n, \mathcal{J}^n_M)$$
where \( S_a(, ) \) is the same function described in Equation (2.12), and \( N \) is the total number of features used in the query and library face graphs. The rest of the recognition procedure shall follow the same process laid out section 5.2.

**5.5. Evaluation of the Combined Recognition Scheme**

This section evaluates the newly proposed Combined Recognition scheme. The evaluation of the Combined Recognition Scheme is done in the same manner as the evaluation of the Weighted and Jet-Mapped Recognition schemes. The recognition tests are performed on 100 face pairs using the k-fold cross-validation procedure, as was done in previous sections.

Training involves calculating the feature weights, as per the procedure in Section 4.2, and generating a suitable jet-mapping using the procedure in Section 5.2. For the testing process, the non-frontal faces in the face pairs are considered to be the query faces. A query face image is selected and using either automatic or manual feature placement operations the feature points are placed and jets are extracted for the face graphs. Jet-mapping is performed on the obtained jets prior to performing the face recognition. During recognition, the weights calculated in training are applied. The recognition rates are then recorded for both manual and automatic feature point placement procedures. The following graph illustrates the some weights generated for the mapped jets. Table 17 tabulates the same numerically.
Figure 38 Plot of Weights Calculated for the different pose graphs, after Jet-Mapping has been applied to the Feature Nodes.

Table 17 Calculation of weights for 15°, 30° and 45° Jet-Mapped Face Pose Graphs for a Sample Set of Faces

<table>
<thead>
<tr>
<th>Node # (n)</th>
<th>Lable</th>
<th>Calculated Weights $W'_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>15° Pose</td>
</tr>
<tr>
<td>1</td>
<td>R.Eb</td>
<td>0.098</td>
</tr>
<tr>
<td>2</td>
<td>L.Eb</td>
<td>0.023</td>
</tr>
<tr>
<td>3</td>
<td>F.C</td>
<td>0.075</td>
</tr>
<tr>
<td>4</td>
<td>E.C</td>
<td>0.102</td>
</tr>
<tr>
<td>5</td>
<td>N.C</td>
<td>0.045</td>
</tr>
<tr>
<td>6</td>
<td>N.Tip</td>
<td>0.055</td>
</tr>
<tr>
<td>7</td>
<td>Cheek</td>
<td>0.031</td>
</tr>
<tr>
<td>8</td>
<td>U.Lip</td>
<td>0.107</td>
</tr>
<tr>
<td>9</td>
<td>L.Corner</td>
<td>0.115</td>
</tr>
<tr>
<td>10</td>
<td>L.Lip</td>
<td>0.099</td>
</tr>
<tr>
<td>11</td>
<td>L.E.Corner</td>
<td>0.125</td>
</tr>
<tr>
<td>12</td>
<td>R.E.Corner</td>
<td>0.124</td>
</tr>
</tbody>
</table>
The tables that follow record the results from the test runs. Tables 18-21 tabulate the recognition results obtained from the test for the Combined Recognition scheme that uses manually placed feature points. Tables 22-24 tabulate the same when using automatic feature point placement scheme described in Chapter 3. The amount of improvement in the recognition rates for both the manual as well as automatic feature point placement schemes is as high as 20% for the larger face pose angles. Even for the smaller face pose angles such as 15°, the recognition scheme introduced in this section has been able to eliminate the drop in recognition due to face pose variation, which is seen in the case of the baseline Direct Recognition scheme.

**Table 18** Evaluation Results for Jet-Mapped Face Recognition scheme using Manually Positioned Feature Points for 15° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct # (%)</td>
<td>Combined # (%)</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td>19 (95%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td>19 (95%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td>20 (100%)</td>
<td>20 (100%)</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td></td>
<td>98%</td>
<td>100%</td>
</tr>
</tbody>
</table>
### Table 19 Evaluation Results for Combined Recognition scheme using Manually Positioned Feature Points for 30° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Direct # (%)</th>
<th>Combined # (%)</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td></td>
<td>10 (50%)</td>
<td>14 (70%)</td>
<td>5%</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td></td>
<td>16 (80%)</td>
<td>18 (90%)</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td></td>
<td>18 (90%)</td>
<td>20 (100%)</td>
<td>0%</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td></td>
<td>13 (65%)</td>
<td>17 (85%)</td>
<td>5%</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td></td>
<td>18 (90%)</td>
<td>20 (100%)</td>
<td>0%</td>
</tr>
<tr>
<td><strong>AVERAGE</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>75%</strong></td>
<td><strong>89%</strong></td>
<td><strong>14%</strong></td>
</tr>
</tbody>
</table>

### Table 20 Evaluation Results for Jet-Mapped Face Recognition scheme using Manually Positioned Feature Points for 45° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Direct # (%)</th>
<th>Combined # (%)</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td></td>
<td>7 (35%)</td>
<td>10 (50%)</td>
<td>15%</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td></td>
<td>12 (60%)</td>
<td>16 (80%)</td>
<td>20%</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td></td>
<td>11 (55%)</td>
<td>17 (85%)</td>
<td>30%</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td></td>
<td>11 (55%)</td>
<td>15 (75%)</td>
<td>20%</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td></td>
<td>15 (75%)</td>
<td>16 (80%)</td>
<td>5%</td>
</tr>
<tr>
<td><strong>AVERAGE</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>56%</strong></td>
<td><strong>74%</strong></td>
<td><strong>18%</strong></td>
</tr>
</tbody>
</table>

### Table 21 Evaluation Results for Jet-Mapped Face Recognition scheme using Automatic Feature Point Positioning for 15° Face Pose Query Images

<table>
<thead>
<tr>
<th>Run Number</th>
<th># of Test Faces</th>
<th># of Training Faces</th>
<th>Face Recognition</th>
<th>Direct # (%)</th>
<th>Combined # (%)</th>
<th>Improvement %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>80</td>
<td></td>
<td>17 (85%)</td>
<td>20 (100%)</td>
<td>15%</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>80</td>
<td></td>
<td>20 (100%)</td>
<td>20 (100%)</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>80</td>
<td></td>
<td>20 (100%)</td>
<td>20 (100%)</td>
<td>0%</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>80</td>
<td></td>
<td>19 (95%)</td>
<td>20 (100%)</td>
<td>5%</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>80</td>
<td></td>
<td>20 (100%)</td>
<td>20 (100%)</td>
<td>0%</td>
</tr>
<tr>
<td><strong>AVERAGE</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>96%</strong></td>
<td><strong>100%</strong></td>
<td><strong>4%</strong></td>
</tr>
</tbody>
</table>
The tables indicate that the Combined Recognition scheme outperforms all the other schemes with manual feature point placement procedure. However, it is more important to note that with automatic feature point placement procedure, which usually degrades performance compared to any manual placement scheme, the Combined Recognition scheme outperforms the initial Direct Recognition scheme with manual placement. This can be better noted in Figure 39 by observing the graph that plots the results obtained in the tables.
Figure 39 Recognition Rates for the Combined Recognition scheme.

The next chapter provides concluding remarks on the different procedures introduced by this thesis and draws out remarks about possible future work.
Chapter 6   Conclusions and Future Work

This thesis introduced a few different approaches to improving face recognition when using Gabor-wavelet based feature descriptor jets. This chapter provides concluding remarks of the methods presented in this thesis, and provides some suggestions on future work.

6.1. Summary of works

6.1.1 Automatic Feature Point Positioning

Automatic positioning of feature point nodes is not accurate enough for high recognition results. This thesis has made an effort to improve automatic feature point positioning using techniques from the Elastic Bunch Graph Matching and Active Shape Models. The results obtained during the evaluation of this process indicate that positioning the feature points is comparable to manual placement of the points.

6.1.2 Feature-Weighted Recognition Scheme

The Weighted Recognition scheme implemented the concept of feature weighting based on their ability to discriminate. The improvement in recognition rates of this scheme over the base recognition scheme has shown that the concept utilized is practical and would be beneficial to implement in recognition systems with hardly any runtime overhead.

6.1.3 Face Derotation Scheme

The Face Derotation scheme was meant to be a practical replacement to high-end 3D operations on an object captured in a 2D image frame. Although image skewing can
sometimes mimic 3D operations on an object in an image frame, it provides acceptable results only for simple object structures.

The implementation of Face Derotation attempted to perform a 2D image transformation that could resemble ‘turning a face’ in an image and generate its frontal view from a non-frontal pose view. Due to complexity in the human face structure, this process would only work for small pose variations within 22°, which was the maximum pose angle used for evaluation of the process. The results obtained show that the procedure is not yet mature enough to handle face images with significant pose variations.

It is preferred that the face database, used to perform the evaluation of a proposed algorithm, demonstrate a desired pattern in a very controlled manner. In the case of this algorithm, the desired pattern is a controlled face pose variation along exactly one axis. However, most faces in the database showed a degree of uncontrolled variations along other pose axes. If a database of faces without these uncontrolled variations were available, the evaluation of the algorithm would be more appropriate. Furthermore, there is a possible solution that can be proposed to this scheme that could improve the performance over uncontrolled variation. This is discussed briefly as possible future work.

6.1.4 Jet-Mapped Recognition Scheme

The Jet-Mapped Recognition scheme was another novel concept for improving the correlation between query feature descriptors and model library feature descriptors. By estimating the mean and standard deviation shift between distributions of the query image feature descriptors and the model library feature descriptors, this scheme established a
mapping of the jet coefficients from a given query face pose image to the frontal face pose image. This mapping may be considered as an alternative to the derotation process, as the mapping tries to negate the effect of face pose variation, much like the Face Derotation scheme. However, this scheme has been more adaptive towards uncontrolled pose variations in the query face. This can be attributed to the training performed specifically for each set of test faces. Hence, given the appropriate training, this recognition scheme is capable of showing a good improvement over the base recognition scheme.

6.1.5 A Combined Recognition Scheme

From the evaluation of Jet-Mapped and Weighted Recognition schemes, it is noted that the improvement in recognition rate for the Jet-Mapped scheme keeps improving for larger face pose angles, as opposed to the Weighted Recognition scheme, which shows decreased improvement for larger pose angles. This indicates that the Jet-Mapped Recognition scheme modifies feature descriptors and is actually geared towards negating the face pose variation, while the Weighted Recognition scheme tries to improve recognition with the available feature descriptors.

Due to the difference in the approach of these two schemes, it has been possible to combine the effects of both to get better results than either of the schemes individually. This result further verifies the fundamentals of both scheme and acts as a supporting ground for further development into either of the schemes. Furthermore, the Combined Recognition scheme has shown the best improvement in recognition results over the base manual Direct Recognition scheme, even with Automatic Feature Point Placement.
6.2. **Scope for Possible Future Work**

6.2.1 **Training for Face Derotation**

The most important improvement that can possibly enhance the results obtained from Face Derotation is the addition of the ability to adapt the algorithm to the face set being tested. The current implementation of the algorithm is quite rigid and allows for very little flexibility with regard to the estimation of target control points. If it were possible to use a training set of manually marked frontal faces that relate to the test face set, then the face graph used as a basis for estimating the positions of the target control points can also be made to relate to the test face set, possibly resulting in a better derotation mapping.

6.2.2 **Improving Automatic Feature Point Placement**

Automatic feature point positioning is a big issue for any recognition algorithm. This work has examined combining the efforts of some object detection schemes in trying to get a good positioning process. With a lot more training, the positioning scheme would probably perform better than the results currently shown. Trying to get the best from the placement procedures can be considered as a possible future work.

6.2.3 **Real-Time Implementation**

The algorithms developed here have been geared towards low overheads for future real-time implementation. The automatic placement procedure has also been primarily developed towards a system where obtaining basic initialization conditions such as eyes and mouth can be achieved by using Harr cascades on an image frame. Each of the recognition schemes has been targeted towards a real-time implementation, during the
course of their development. The development of a real-time implementation of the work presented in this thesis could be the focus of future work in this area.
Bibliography


