Modeling distance functions induced by face recognition algorithms

Soumee Chaudhari

University of South Florida

Follow this and additional works at: http://scholarcommons.usf.edu/etd

Part of the American Studies Commons

Scholar Commons Citation


This Thesis is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
Modelling Distance Functions Induced by Face Recognition Algorithms

by

Soumee Chaudhari

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Sudeep Sarkar, Ph.D.
Rangachar Kasturi, Ph.D.
Ravi Sankar, Ph.D.

Date of Approval:
November 9, 2004

Keywords: affine space, eigen space, principal component analysis, optimal affine transformation, biometrics

© Copyright 2004, Soumee Chaudhari
DEDICATION

To my family and fiance who have been a constant source of love and support
ACKNOWLEDGEMENTS

I would like to thank Dr. Sudeep Sarkar, my major professor, for his guidance and support throughout my Master’s degree program, and for carefully reviewing my thesis writeup. This thesis has been a great learning experience for me. I sincerely thank Dr. Sudeep Sarkar for giving me the opportunity to work on this project. I would also like to thank Dr. Kasturi and Dr. Sankar for being a part of my thesis committee. I would also like to thank Dr. Ross Beveridge and his students from the Colorado State University for their patience and help. I would also like to thank my fellow students Lakshmi Reguna, Pranab Mohanty and Himanshu Vajaria for their valuable and timely help.
# TABLE OF CONTENTS

LIST OF FIGURES iii

ABSTRACT v

CHAPTER 1 INTRODUCTION 1
  1.1 Motivation 3

CHAPTER 2 RELATED WORK 5
  2.1 Face Recognition Evaluation Protocols 5
  2.2 Dimensionality Reduction Techniques 10
  2.3 Distance Metric Learning Techniques 13

CHAPTER 3 THE FACE RECOGNITION ALGORITHMS EVALUATED 15
  3.1 Principal Component Analysis 16
  3.2 Linear Discriminant Analysis 16
  3.3 Bayesian Intrapersonal/Extrapersonal Classifier 17
  3.4 Elastic Bunch Graph Matching 18
  3.5 Commercial Face Recognition Software 19

CHAPTER 4 DISTANCES IN AFFINE APPROXIMATION SPACES 21
  4.1 Multi-Dimensional Scaling 21
  4.2 Affine Transformation Based Modelling of Distances 22
    4.2.1 Similarities and Dissimilarities 25
  4.3 An Alternative Method to Derive A 25

CHAPTER 5 EXPERIMENTS AND ANALYSIS OF RESULTS 27
  5.1 Issues Addressed 27
  5.2 Distance Measures Used by Face Recognition Algorithms 27
  5.3 Data Description 29
  5.4 Training and Test Setup 31
  5.5 Affine Matrix 34
    5.5.1 Affine Space Dimensions 34
  5.6 Performance on Data with Time Variation 40
    5.6.1 Identification and Verification Performance 40
  5.7 Performance on Data With Expression Variation 42
    5.7.1 Identification and Verification Performance 42
  5.8 Approximating PCA with Different Distance Measures 44
  5.9 Effects of Normalization on the Distance Matrix 45
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Evaluation Protocols</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Dimensionality Reduction Techniques</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Distance Metric Learning Techniques</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>Face Recognition Algorithm</td>
<td>15</td>
</tr>
<tr>
<td>4.1</td>
<td>Find Matrix $A$ such that the Euclidean Distances Between Transformed images are Equal to the Given Distances</td>
<td>21</td>
</tr>
<tr>
<td>5.1</td>
<td>Training Set: 100 Images with 4 Images of Each Subject of the FERET Data Set</td>
<td>30</td>
</tr>
<tr>
<td>5.2</td>
<td>Test Set(FERET): 600 Images with 2 Images of Each Subject</td>
<td>31</td>
</tr>
<tr>
<td>5.3</td>
<td>Test Set(Notre Dame): 830 Images with 2 Images of Each Subject</td>
<td>31</td>
</tr>
<tr>
<td>5.4</td>
<td>Histogram of the Notre Dame Images Acquired Over Time</td>
<td>32</td>
</tr>
<tr>
<td>5.5</td>
<td>Sample Images</td>
<td>32</td>
</tr>
<tr>
<td>5.6</td>
<td>Training Setup for the Affine Approximation algorithm</td>
<td>33</td>
</tr>
<tr>
<td>5.7</td>
<td>Testing Setup for the Affine Approximation algorithm</td>
<td>33</td>
</tr>
<tr>
<td>5.8</td>
<td>Visualization of $A_{nr}^T A_{nr}$ for LDA Algorithm</td>
<td>35</td>
</tr>
<tr>
<td>5.9</td>
<td>Visualization of $A_{nr}^T A_{nr}$ for BIC Algorithm</td>
<td>35</td>
</tr>
<tr>
<td>5.10</td>
<td>Visualization of $A_{nr}^T A_{nr}$ for EBGM Algorithm</td>
<td>36</td>
</tr>
<tr>
<td>5.11</td>
<td>Visualization of $A_{nr}^T A_{nr}$ for Commercial Algorithm</td>
<td>36</td>
</tr>
<tr>
<td>5.12</td>
<td>Visualization of Diagonal Values of $A_{nr}^T A_{nr}$ for All Algorithms</td>
<td>37</td>
</tr>
<tr>
<td>5.13</td>
<td>Top Dimensions of the Affine Approximation to the Different Algorithms. Last Row Shows the Corresponding PCA Dimension for Comparison</td>
<td>38</td>
</tr>
<tr>
<td>5.14</td>
<td>PCA Eigen Dimensions Along Which We Need to Stretch and Shear the Most to Match Distances</td>
<td>39</td>
</tr>
</tbody>
</table>
Figure 5.15 ROC on Notre Dame Images Showing the Performance Comparison of the Different Face Recognition Algorithm along with the Affine Approximation Algorithm (a) LDA (b) BIC (c) EBGM (d) Commercial

Figure 5.16 CMC on Notre Dame Images Showing the Performance Comparison of the Different Face Recognition Algorithm along with the Affine Approximation Algorithm (a) LDA (b) BIC (c) EBGM (d) Commercial

Figure 5.17 ROC on FERET Data Set Showing the Performance Comparison of the Different Face Recognition Algorithm along with the Affine Approximation Algorithm. (a) LDA (b) BIC (c) EBGM (d) Commercial

Figure 5.18 CMC on FERET Data Set Showing the Performance Comparison of the Different Face Recognition Algorithm along with the Affine Approximation Algorithm (a) LDA (b) BIC (c) EBGM (d) Commercial

Figure 5.19 ROC Curves of PCA Algorithm with Different Distance Measures (a) Euclidean (b) Covariance (c) Mah-Cosine (d) MahL1

Figure 5.20 ROC of PCA Algorithm with Different Distance Measures (a) Euclidean (b) Covariance (c) Mah-Cosine (d) MahL1

Figure 5.21 Visualization of $A_n^T A_{nr}$ for PCA Algorithm (Euclidean Distance) for the FERET Data Set

Figure 5.22 Visualization of $A_n^T A_{nr}$ for PCA Algorithm (Covariance Distance) for the FERET Data Set

Figure 5.23 Visualization of $A_n^T A_{nr}$ for PCA Algorithm (MahCosine Distance) for the FERET Data Set

Figure 5.24 Visualization of $A_n^T A_{nr}$ for PCA Algorithm (MahL1 Distance) for the FERET Data Set

Figure 5.25 Effect of Normalization on the ROC Curves for the Commercial Face Recognition Algorithm on the FERET Data Set

Figure A.1 Comparison of CMC Using FERET Data on Different Algorithms

Figure A.2 Comparison of ROC Using FERET Data on Different Algorithms
Modelling Distance Functions Induced by Face Recognition Algorithms

Soumee Chaudhari

ABSTRACT

Face recognition algorithms have in the past few years become a very active area of research in the fields of computer vision, image processing, and cognitive psychology. This has spawned various algorithms of different complexities. The concept of principal component analysis (PCA) is a popular mode of face recognition algorithm and has often been used to benchmark other face recognition algorithms for identification and verification scenarios. However in this thesis, we try to analyze different face recognition algorithms at a deeper level. The objective is to model the distances output by any face recognition algorithm as a function of the input images. We achieve this by creating an affine eigen space from the PCA space such that it can approximate the results of the face recognition algorithm under consideration as closely as possible.

Holistic template matching algorithms like the Linear Discriminant Analysis algorithm (LDA), the Bayesian Intrapersonal/Extrapersonal classifier (BIC), as well as local feature based algorithms like the Elastic Bunch Graph Matching algorithm (EBGM) and a commercial face recognition algorithm are selected for our experiments. We experiment on two different data sets, the FERET data set and the Notre Dame data set. The FERET data set consists of images of subjects with variation in both time and expression. The Notre Dame data set consists of images of subjects with variation in time. We train our affine approximation algorithm on 25 subjects and test with 300 subjects from the FERET data set and 415 subjects from the Notre Dame data set. We also analyze the effect of different distance metrics used by the face recognition algorithm on the accuracy of the approximation. We study the quality of the approximation in the context of recognition.
for the identification and verification scenarios, characterized by cumulative match score curves (CMC) and receiver operator curves (ROC), respectively.

Our studies indicate that both the holistic template matching algorithms as well as feature based algorithms can be well approximated. We also find the affine approximation training can be generalized across covariates. For the data with time variation, we find that the rank order of approximation performance is BIC, LDA, EBGM, and commercial. For the data with expression variation, the rank order is LDA, BIC, commercial, and EBGM. Experiments to approximate PCA with distance measures other than Euclidean also performed very well. PCA+Euclidean distance is best approximated followed by PCA+MahL1, PCA+MahCosine, and PCA+Covariance.
Recently biometric based authentication and identification systems have received a great impetus. They have found numerous applications in surveillance, secure site access, transaction security and remote access to resources. But unlike non biometric systems they have several inherent advantages. Unlike a pin or a password it cannot be used by an unauthorized user, it does not need to be carried, and provides for positive identification. However biometric based fingerprint, iris or DNA recognition are intrusive and require the cooperation of the user. This is one the major factors why face recognition has become highly popular and an active area of research. It has the advantage of being non invasive and can be performed even at a distance, some times even without the knowledge of the user as is necessary in many security and surveillance applications.

Face recognition technology can be done with 2D images as well as, as the recent trend has been, with 3D images. Algorithms to perform frontal, profile and view tolerant recognition have been developed. Being a challenging yet interesting problem, various algorithms and techniques have been developed for face recognition problems. It can be broadly classified into three techniques. Holistic methods, feature-based methods and hybrid methods. Holistic methods include methods such as principal component analysis [1], linear discriminant analysis [2], and support vector machines [3]. Feature-based methods include pure geometry methods, dynamic link architecture [4], and hidden Markov models [5]. Hybrid methods include modular eigen faces, hybrid LFA, and component based methods [6]. The process may include automatically detecting or locating the face in the image, extraction of the orientation of the face, handling of conditions such as illumination and shadows and generation of new views from existing views.
In the wake of the recent terrorism incidents, especially after 9/11 there has been renewed interest in face recognition technology. However it has come in for a lot of criticism for its failure to perform adequately in various airports and other sensitive locations, like border check points that it has been installed. Several other government agencies have abandoned face recognition systems after finding that their performance was not close to what was advertised. Since most face recognition systems are easily thrown off by changes in hairstyle, facial hair, aging, weight gain or loss and minor disguises we need more research and effective algorithms to deal with these issues. Consequently we also need effective protocols to evaluate these algorithms so as to maintain a standard of performance.

Two major evaluation methodologies used recently were the FERET \[7, 8, 9, 10, 11\] and the FRVT \[12, 13, 14\] evaluation protocols. The FERET testing model evaluated the algorithms based on different scenarios, different categories of images and versions of algorithms. Performance was computed under two scenarios. The identification scenario and the verification scenario. In an identification application, the algorithm is presented with an unknown face which needs to be identified from a set of images. In the verification application, the algorithm needs to determine the subject in the image is indeed who it claims to be from a set of images. The FERET evaluations took place in 1994, 1995 and 1996. FRVT 2000 and 2002 evaluations measured the capabilities of systems on real life large scale databases and introduced new experiments to understand face recognition experiments better. The size, complexity and difficulty level was increased with each successive evaluation in order to reflect the increasing maturation of the face recognition technology as well as evaluation theory.

The technological evaluation of these face recognition algorithms is carried out by generation of empirical statistics of their performance for the identification and verification scenarios. However these statistics are dependent on the size and characteristics of the database and offer only a global (or gross) characterization of the algorithms. Our aim in this thesis has been to evaluate the face recognition algorithms at a deeper level. We try to approximate the performance of any algorithm by an approach that just performs affine transformation (rotation, shear, stretch) of the image space.
1.1 Motivation

At the core of any face recognition algorithm is a module that computes distance (or similarity) between two face images. Just as linear systems theory allows us to characterize a system based on inputs and outputs, we seek to characterize a face recognition algorithm based on the distances (the “outputs”) computed between two faces (the “inputs”). Can we model the distances, \(d_{ij}\), computed by any given face recognition algorithm, as a function of the given face images, \(\vec{x}_i \rightarrow \phi(\vec{x}_i)\) and \(\vec{x}_j \rightarrow \phi(\vec{x}_j)\), where \(\vec{x}_i\) and \(\vec{x}_j\) are the row scanned vector representations of the given image array? Mathematically, what is the function \(\phi\) such that \(d_{ij} - ||\phi(\vec{x}_i) - \phi(\vec{x}_j)||\) is minimized? The benefits of such as characterization is twofold. First, it will allow us to compare any two recognition algorithms at a deeper level than is presently possible, using just overall performance scores, that are dependent on the size and characteristics of the database. For instance, if \(\phi\) is an orthogonal operator then it would suggest that the underlying face recognition algorithms is essentially performing a rigid rotation and translation to the face representations, i.e. like in principal component analysis (PCA). If \(\phi\) is a general affine operator, then it would suggest the underlying algorithms can be approximated fairly well by a linear transformation (rotation, shear, stretch) of the face representations. The second benefit is that the proposed characterization will facilitate future analytical modelling of complete, possibly multi-modal, biometric systems, where the recognition module would be a part of network of different modules. A function representation of the face recognition module will allow us to analytically express variations in the output (the distances) in terms of the variations in the input images.

In this thesis we specifically consider affine \(\phi\)'s and derive a closed form solution that use the statistical method of multi-dimensional scaling (MDS) [15]. Given a distance matrix between a set of data, MDS can arrive at an embedding of the data in a multi-dimensional space, where the distances are close to the given distance matrix. However, unlike MDS, which provides an embedding just for the given data set, we learn a function that can be used for new data points. We analyze some of the popular face recognition algorithms, namely Eigenfaces (PCA + distance metrics) [1], Linear Discriminant Analysis (LDA) [2], Bayesian Intrapersonal/Extrapersonal Classifier (BIC) [16], and Elastic Bunch
Graph Matching (EBGM) face recognition algorithm [4] and a commercial face recognition algorithm. The choice of the face recognition algorithms span template based approaches to feature based ones, such as the EGBM and commercial algorithm. We test the quality of the approximation on the well known FERET [17] data set and on the recent Notre Dame data [18]. We measure the approximation quality by the recognition rates using the learnt affine models. Surprisingly, the results, which is based on a complete separation of train and test data and on large test data sets, indicates that the algorithms can be fairly well approximated by affine transformation models.

This study is a follow up on the work done by Lakshmi Reguna in her Masters thesis [19]. The current thesis is different from Reguna’s in the following ways,

1. We have a new approximation strategy which is significantly faster than Reguna’s.

2. We experiment and test on more algorithms including template matching algorithms as well as feature based graph algorithms, than Reguna. Certain experiments which took up more than a day to execute now take just few minutes with our approach.

3. We also test with two different data sets with variation in expression and time.

4. We also have complete separation of train and test set in terms of collection site.

The following chapters of this thesis has been organized in the following way. In the next chapter, we take a look at some of the related work that has been done. Next, we take a brief look at the test face recognition algorithms that we have experimented with. Chapter 4 describes the mathematical formulation behind achieving the affine transformation. Finally, we have described the studies and the results obtained followed by conclusions.
CHAPTER 2
RELATED WORK

This chapter is divided into three main sections. Given the close similarity of goals with our approach, we discuss the dominant evaluation protocols and methodologies that have been used to evaluate the performance of face recognition algorithms. We also discuss various methods and techniques that are being used for dimensionality reduction in cases of high dimensional data. Then we take a look at works on learning distance metric from the context of machine learning.

2.1 Face Recognition Evaluation Protocols

![Diagram of evaluation protocols]

Figure 2.1. Evaluation Protocols

The FERET evaluations [7, 8, 9, 10, 11] are some of the most significant research that has been done to evaluate the performance of face recognition algorithms. It had three pri-
mary purposes. The goal was development and advancement of face recognition technology. The second goal was to create a large image database that could be used as a standard for algorithm development, test and evaluation. The third goal was to provide standardized tests and protocols for face recognition algorithm. The tests measured the performance of the algorithms on various aspects, like its ability to handle large databases, variation in illumination, scale, pose and changes in background. Performance was measured in terms of probability of false alarm and false negatives. A false alarm is the scenario where a face is falsely recognized to exist in the gallery and a false negative is the scenario when a face is not recognized even though it does exist in the gallery. The FERET program ran in phases from the years 1993 to 1996. The first FERET evaluation took place on August 94. The aim of these tests were to evaluate the ability of algorithms to locate, normalize and identify faces from a database. The tests included ability to recognize from a gallery, rate of false alarms and resilience to pose variation. The second evaluation took place on March 1995 and was designed to measure the progress of the algorithms since the first evaluation. They were also tested on larger databases and with duplicate images taken over a period of time. The third and final test was done on September 96 where the performance was tested on over 3000 images and against multiple probe and gallery sets. The gallery and probe sets were selected based on various combinations of variation over time and lighting conditions. The algorithms tested were the PCA [1], Fisher discriminant analysis [2], Greyscale projection [20], Dynamic Link Architecture [4], and Bayesian classification [16] based approaches.

The FERET evaluations examined the performance for both the closed universe as well as the open universe scenarios. The open universe scenario implies that not every probe exists in the gallery. In the closed universe scenario every probe exists in the gallery. The closed universe scenario is also called the identification scenario because it allows us to know if the probe image is the top identifying match or how many images need to be examined to know the get the desired level of performance. They are characterized by Cumulative Match score Curves (CMC) where the x-axis denotes the rank and the y-axis denotes the percentage of correct matches. The higher the percentage of correct matches at rank one the better the recognition rate. In the verification scenario, the
system either confirms or denies if a probe exists in the gallery. It is characterized by the Receiver Operator characteristic Curve (ROC). The x-axis represents the false alarm rate and the y-axis represents the probability of verification. The higher the probability of verification at low false alarms the higher is the accuracy of the system. Identification scenario applications include identifying a face from a set of mug shots or surveillance images at airports. Verification scenario applications include automated confirmation of identities at ATM machines, access control to secure buildings and sites.

Some of the conclusions drawn from the FERET evaluations were.

1. Algorithm performance increased with increase in size of database.

2. Algorithm performance also depended on the selection of probe and gallery sets.

3. Since some algorithms performed better for identification than verification, performance on one task is not predictive of performance on another.

4. Identification performance on duplicate scores (match frontals taken on different dates) was lower than performance on frontals taken on same date. Verification results also had similar conclusions regarding this.

5. Some algorithms were insensitive to change in illumination and some showed performance degradation after 40% change in illumination.

6. Most algorithms were insensitive to change in image size.

Due to the FERET evaluations, it was possible for researchers to develop and test their algorithm across a common database using standard evaluation protocols. It enabled the face recognition community to assess the strengths and weaknesses in the field and lay the foundations for future direction of research.

The Face Recognition Vendor Test (FRVT) [12, 13, 14] conducted in 2000 and 2002 followed the FERET evaluations except that they were conducted on commercially available and mature commercial face recognition systems. FRVT 2000 followed the three step evaluation protocol of technology evaluation, scenario evaluation, and operational evaluation. The Recognition Performance Test (Technology Evaluation) showed how well the
various systems responded to changes in pose, lighting, and image compression level. The Product Usability Test (Limited scenario evaluation) demonstrated the ability of a face recognition system to operate under a live environment. Participants of the FRVT 2000 include Banque-Tec International Pty Limited, C-Vis Computer Vision and Automation, eTrue (formerly Miros), Lau Technologies, and Visionics Corporation.

FRVT 2002 attempted to assess the performance of the algorithms to meet real world requirements. It consisted of two sub-tests, the high computational intensity (HCInt) test and medium computational intensity (MCInt) test. The HCInt test was used to test the performance of the systems on very challenging real-world problems. On the other hand, MCInt was designed to evaluate the performance of systems on various types of images (still as well as video) under various conditions. There were many more participants in the FRVT 2002 including AcSys Biometrics, C-VIS Computer Vision and Automation, Cognitec Systems, Dream Mirh Co., Ltd, Eyematic Interfaces Inc., Iconquest, Identix, Imagis Technologies Inc., Viisage Technology, and VisionSphere Technologies.

Some of the conclusions drawn from the FRVT tests include:

1. Performance decreases linearly with increase in elapsed time between database and new images. Performance dropped roughly at a rate of 0.05 points per year for identification scenario. However the performance drop for verification scenario was slower than for identification.

2. Good face recognition systems are resilient to indoor lighting change.

3. Non frontal faces are better recognized by re-mapping them into frontal ones using 3D morphable models [21].

4. Video images do not necessarily result in better performance than still image.

5. Older mature face are easier to recognize than younger faces.

6. Male faces are easier to recognize than female faces.

7. Outdoor images do not perform as well as indoor images.
There have been several other efforts to benchmark and evaluate face recognition algorithms. Gutta et al. [22] conducted benchmark studies on several simple but well known algorithms as well as novel recognition schemes. Some of the conclusions that they drew of their studies, was that the future of face recognition algorithms lay with hybrid recognition systems and that holistic algorithms outperform feature and correlation methods.

Various efforts were also made to develop evaluation frameworks using statistical evaluation methodologies of recognition algorithms. In [23] the authors have suggested a new framework for evaluating recognition systems. Often, just blind assumption of i.i.d data can reduce the accuracy of a system. Their work allowed the system to obtain tight confidence intervals of evaluation estimates. It also simultaneously reduced the amount of data and computation required to reach those conclusions. They have achieved this using stratified sampling methods and the application of a replicate statistical technique called balanced repeated sampling (BRR). Some parametric and non parametric methods for the statistical evaluation of recognition algorithms were explored in [24]. One of the methods proposed was a parametric method that equated success or failure of algorithms on probe images to Bernoulli trials. This method tries to provide a probability of success of the algorithm arising due to the size of the sample. The second non parametric method based on the Monte Carlo based sampling technique captures the probability distribution of the recognition rate by sampling the space of all possible gallery and probe sets.

An interesting piece of work done by Philips et al. [25] evaluated various face recognition algorithm to assess the qualitative accord between face recognition algorithms and human perceivers. The aim of this research was to answer the question if both humans and algorithms found the same faces similar. Unlike the FERET evaluations it concentrated on the qualitative rather than the quantitative aspects of performance by humans and models. It also tried to use faces that were confused with each other. It concluded that most face recognition algorithms performed similar to humans. By comparing similarity scores between pairs of faces and measure generated by algorithmic models and humans. It was also observed that algorithms with different representations could cluster together in terms of distance computed. Most algorithms with same representations cluster together
although it was not the only factor. Another important observation was that algorithms having the same representations but using different distance metrics clustered differently.

2.2 Dimensionality Reduction Techniques

Interest in visualization of high dimensional data face the problem of dimensionality reduction. Being able to find meaningful lower dimensions to represent the data hidden in their high dimensional observations has been a challenge faced by many fields of interest. The goal is to estimate low dimensional placement of a given set of points so as to approximate distances in a higher dimensional space. In this aspect our goal is similar to Multidimensional Scaling (MDS) [15]. However we also seek to learn a mapping so that we can map new data points into this space. Some of the dimensionality reducing techniques are shown in Fig. 2.2.

![Dimensionality Reduction Techniques](image)

**Figure 2.2. Dimensionality Reduction Techniques**

Embedding one space into a lower dimensional space has generally employed linear techniques like PCA or nonlinear techniques like MDS. Embedding implies the mapping of one space into another. In PCA the optimal “m” dimensions are retained. The chosen coordinate axes coincide with the eigenvectors with the large eigenvalues. Euclidean distances in this reduced subspace approximate the Euclidean distances in the original space. In MDS,
a low dimensional space is created in which the dissimilarity or similarity between any two objects is preserved. However both these methods become computationally expensive for high dimensional data or if the number of points is large. Also PCA is typically suitable for data that has simple linear correlations. MDS on the other hand, is an iterative process except for the simplest case and does not guarantee optimality or uniqueness in output. Newer methods like manifold learning methods which include the Isomap [26] and LLE [27] algorithms use a collection of local neighborhoods or exploit the spectral properties of adjacency graphs from which the global geometry of the manifold can be reconstructed. A distributional scaling method by [28] describes a new method for embedding metric as well as non metric spaces in low dimensional Euclidean spaces. This method works with metric as well as non metric data sets. The method combines both pairwise distortion as well as geometric distortion. It tries to preserve the original structure or geometry of the data. It is also resilient to the presence of noise. It uses clustering algorithms to direct the embedding process and improve its convergence properties. They also suggest methods to estimate the right dimensionality of data. These methods are based on a local geometric approach and a global heuristic approach.

The theory of multidimensional scaling [15](MDS) has been used in the past mostly in relation to face recognition as a visualization tool for distances between faces. MDS can be defined as a search for a low dimensional Euclidean space in which the distance between the points in space match as well as possible to the original dissimilarities. In our study multidimensional scaling is the theory behind the Affine Approximation Algorithm. From a set of known squared distances $d_{rs}$ in the original space, we calculate the inner product matrix $B$ and from $B$ the coordinates in the reduced space. The objective is to find coordinates such that the distance between the points in the reduced space "matches" as well as possible the distances in the original space. The dissimilarities/distances taken as input in practical cases may or may not be Euclidean. If these dissimilarities are not Euclidean the $B$ matrix will not be positive semi-definite which is indicated by some negative eigenvalues. To make it positive semi-definite, a constant can be added to all the dissimilarities except the self dissimilarities [15].

$$\delta'_{rs} = \delta_{rs} + c(1 - \delta_{rs})$$
where $c = -2\lambda_n$ is a constant and $\delta_{rs}$ is Kronecker delta. $\lambda_n$ is the smallest eigenvalue.

A non-linear dimensionality reduction method described in [26] uses the algorithmic benefits of PCA and multidimensional scaling (MDS) but learns a broad class of non-linear manifolds. Data which lies on a two-dimensional Swiss roll have points lying far apart as measured by geodesic distances on an underlying manifold may appear to be close to each other if they are measured by Euclidean distances. The geodesic distances between far-away points given the input space distances can be measured in Euclidean or any other domain specific metric. For near or neighboring points the input space distances gives a good estimate of the geodesic distances. Geodesic distances for far-away points are approximated by finding the shortest path between them, found by constructing a graph connecting neighboring points. The first step in this method, called Isomap, entails estimating neighboring points on a manifold based on a fixed radius or k nearest neighbor methods. The second step estimates the geodesic distances between all pairs of input points by taking the shortest path between them. The final step is to apply classical MDS to the matrix of graph distances. A d-dimensional Euclidean space is obtained which best preserves the manifolds intrinsic geometry. It is capable of discovering nonlinear degrees of freedom unlike the classical techniques of PCA and MDS.

Other techniques of nonlinear dimensionality reduction include the local linear method (LLE) as described in [27]. It is an unsupervised algorithm that computes low dimensional, neighborhood preserving, embedding of high dimensional inputs. Unlike other clustering methods for local dimensionality reduction, LLE maps its input to a low dimensional global coordinate system and does not get stuck in a local minima problem. This method involves taking a sample set of data points representing the underlying manifold. The underlying geometry of these patches is reconstructed from the linear coefficients that reconstruct each data point from its neighbors. A cost function represents the reconstruction error which is then minimized to get the optimal reconstruction. This method scales well with manifold dimensionality. It avoids the problem of solving large dynamic programming problems and uses sparse matrices that can be exploited to save computation time and space. These methods are powerful, non-iterative, and guarantee global optimality. However they do not perform well if the number of data points is less or if the data is intrinsically non-metric.
A new method for dimensionality reduction or learning underlying manifolds was proposed in [29], based on semi definite programming. It combines the concepts of semi-definite programming for learning kernel matrices with spectral methods of non linear dimensionality reduction. Like the Isomap and the LLE algorithms it is not affected by local minima problems. The first step of this algorithm includes finding the k nearest neighbor of each input and create a graph linking each neighbor as well as each neighbor to other neighbors of the same input. The Gram matrix of the maximum variance embedding is computed which is centered at the origin and preserves the distances of all edges in the neighborhood graph. The final step extracts the low dimensional embedding from the dominant eigenvectors of the Gram matrix learned by semi-definite programming.

2.3 Distance Metric Learning Techniques

Several fields like artificial intelligence, pattern classification, machine learning, statistics data analysis require us to be able to learn important information hidden in multivariate data. Of particular interest are works that seek to learn distances for examples of similar and non similar classes.

Figure 2.3. Distance Metric Learning Techniques
In [30] a distance metric learning method is proposed that learns a distance metric that seeks to preserve the similarity/dissimilarity (binary 0 or 1) relationship between a set of points. The method is based on posing distance metric learning problem as a convex optimization problem. This method scored over MDS and other distance metric learning algorithms in that it can learn the metric over the whole input space and not just the training points. Hence it performs better with previously unseen data. This method is also efficient and local optima free. However this method involves an iterative procedure and eigen decomposition and can become expensive with large number of dimensions.

A distance metric learning algorithm with kernels was proposed in [31]. It describes a feature weighting method that worked in the input space as well as the kernel space. It basically performs a non parametric kernel adaptation. Many clustering or classification algorithms make use of a distance measure between patterns. One of the most popular method being the Euclidean distance metric. However the Euclidean distance assumes equal weightage to all dimensions. In real world applications this is rarely true. Hence feature weighting techniques are used. But the limitation is that the number of parameters or weights increases with the increase in number of features. Hence they cannot be easily kernelized and can typically select features only in the input space and not in the feature space.

The distance learning method described in [32] learns by relative comparisons, which is a flexible way for describing qualitative training data as a set of constraints. These constraints lead to a convex quadratic programming problem that can be solved by adapting standard methods for SVM training. It can learn a distance metric from qualitative and relative examples. The algorithm searches a parameterized family of distance metrics and discriminately searches for the parameters that fulfill the training examples. A training problem is then formulated as a convex quadratic program for learning the weights associated with these dimensions. The advantage of this particular algorithm is that the qualitative nature of the constraint enables it to be used for a wide range of applications.
For our experiments five face recognition algorithms were selected. The Colorado State University source implementation [33] was used to get the output distance matrices for the PCA, LDA [34], BIC [35] and EBGM [36] algorithms. A top performing commercial face recognition software was the fifth algorithm tested. As shown in Fig. 3 we have selected algorithm based on both holistic template matching techniques as well as local feature based graph algorithms. The PCA, LDA and BIC algorithms are the template matching algorithms and local feature based graph algorithms include the EBGM and the commercial face recognition algorithm.

![Figure 3.1. Face Recognition Algorithm](image)
3.1 Principal Component Analysis

This theory is motivated by both physiology and information theory. It recognizes faces based on the fact that faces can be recognized based on certain set of image features which approximate the image face. However these features do not necessarily correspond to the intuitive notion of facial features. In mathematical terms it can be said that we need to find the principal components of the distribution of the input data set which can be shown to be the principal eigenvectors of the covariance matrix of the input images. These eigenvalues and corresponding eigenvectors are then ordered. Each eigenvalue corresponds to a certain amount of variation along each dimension. We can choose to keep the top M eigenvectors that best describe the variance in the images. Each image can now be represented in a linear combination of the projections on these top eigenvectors. Recognition of new faces is performed by projecting this face in a new eigenspace and then comparing the distance of this face from the faces in the training set. If the lowest distance is within a certain amount, it is considered to be that individual else it is classified as unknown. In a real world application, the trade off between accuracy and speed needs to be assessed. Accuracy increases with the number of faces in the database, however this also decreases the speed of recognition. A few issues need to be taken care of before a system can successfully work. Background information can have a significant effect on the performance since the eigenvectors are calculated on the whole image and not just the face image. Hence elimination of the background is an important prerequisite for good performance. A second consideration is that the performance decreases if the scale/size of the faces is not close to the trained images. The orientation of the head also has an effect on performance. Orientation of the head along the lines of the eigen faces benefits the recognition rate greatly. Recognition rates fall off as the head orientation differs from that in the training set of images.

3.2 Linear Discriminant Analysis

This algorithm is a combination of PCA and LDA. PCA is used as the preliminary step to create a face subspace to apply LDA so as to obtain the best linear classifier. The
combination also helps in classification when there are few number of samples from each class available. This is because a pure LDA algorithm is finely tuned to the training data and does not generalize well. Hence by giving only the principal components as the input to the classifier, better generalization is achieved. The goal of this classifier is to reduce the within class scatter and increase the between class scatter and form a subspace that linearly separates between classes. The procedure includes mapping an image $x$ onto the face subspace $\Theta$ to get $y$. This in turn is projected on the discriminant space $W_y$ to get $z$.

$$\bar{y} = \Theta(x - \bar{m})$$

$$\bar{z} = W_y^T \bar{y}$$

$x$ is the mean image. Finally, classification is performed based on some distance metric like the Euclidean distance. Performance using this hybrid method was show to have improved over a pure LDA classifier.

### 3.3 Bayesian Intrapersonal/Extrapersonal Classifier

This algorithm proposed by Moghaddam and Pentland [35] uses the difference between two images to probabilistically determine whether they belong to the same subject or not. Difference images arising from the images of the same subject are called intra-personal images and difference images arising from the images of two different subjects is called as extra-personal images. Each of these difference images is considered to be a point in a high dimensional space. The high dimensional space is however very sparsely populated as majority of the vacant spaces correspond to difference images that never occur in practice. These difference images will tend to form clusters. Moghaddam and Pentland assume that each difference image belongs to one of the two interpersonal and extra personal clusters. Also that they are distinct and localized Gaussian distributions within the space of all possible images. However the parameters to these distributions are unknown. These parameters can be estimated by using the maximum a posteriori method or the maximum likelihood method. For this thesis we will restrict ourselves to only the Maximum Likelihood
method since it has been found to have equally good results as the Maximum a posteriori method and the same time is much less computationally intensive.

In the training phase, PCA is performed to estimate the statistical properties of the two subspaces i.e. the interpersonal class called $\Omega_I$ and extra personal class $\Omega_E$. If $\Delta$ is a difference image of unknown membership then the similarity score for the maximum likelihood is given as

$$S_{ML} = P(\Delta|\Omega_I)$$

$$P(\Delta|\Omega_I) = \frac{e^{-0.5 \sum_{t=1}^{T} \frac{y_t^2}{\lambda_t}}}{(2\pi)^{T/2} \prod_{t=1}^{T} \lambda_t^{1/2}}$$

where $T$ - no of truncated dimensions from the original dimensions of the data (in order to reduce computational complexity)

$\lambda$ - $T$ eigen values $y = [y_1, y_2, \cdots, y_M]^T$ of each difference image $\Delta$ embedded into the PCA subspace.

During the testing phase the classifier takes a difference image of unknown membership and uses $P(\Delta/\Omega_I)$ as a means of identification. The maximum likelihood estimate ignores the extra-personal class information. When comparing a novel image to $n$ known gallery images, the gallery image yielding the highest similarity score is taken as to be the person in the probe image.

### 3.4 Elastic Bunch Graph Matching

This algorithm differs from the other algorithms because it recognizes faces by comparing parts, instead of performing matching the whole image. The features of the images are represented by Gabor jets also called as model jets. This is obtained by convolving an image with Gabor filters. The model jets are collectively called bunch graphs. Each node in this graph is a collection of model jets of a particular landmark. These jets have been extracted from manually selected landmark locations from the model images and adding to the appropriate bunch. These bunch graphs are then used as reference data for landmark descriptions while locating landmarks in novel images.
Locating a landmark is based on two steps. The location is first estimated by the known location of other landmarks in the image. This estimated location is then further refined by extracting a Gabor jet at that point and comparing it against a set of models. The most similar jet is selected from a bunch graph and this then serves as a model. The algorithm begins by estimating the eye coordinates first because these estimates are very reliable. The algorithm works iteratively to locate the rest of the landmarks till it has reached the edge of the head.

Face graphs are created for each image by extracting jets from the landmark locations like eyes, nose tip, and corner of lips. These graphs contain the physical location of the landmarks as well as the value of the jets. Jets are also extracted from locations at the midpoint between two landmarks. Since an image is represented only by its face graph now, the original image data can be discarded.

Similarity between two images is calculated as a function of the landmark locations and their jet values. Jet similarity can be computed using various methods of magnitude only, phase or displacement compensated Gabor jet similarity. Another method to compute the similarity is based on the position of the landmark points. A simple way is to compute the Euclidean distance between these locations. The presumption being that images belonging to the same subject will differ very little in the landmark locations.

3.5 Commercial Face Recognition Software

A commercial face recognition system was chosen for our experiments. This particular system was amongst the top performing algorithm in all the FRVT evaluations. It is capable of capturing images at a distance and in motion using CCTV and can perform real time identification of subjects. It uses Local Feature Analysis (LFA) to represent the face. The mathematical technique of LFA assumes that a facial image can be synthesized from an irreducible set of building elements. These elements can be derived from a set of model face images using statistical techniques. For identification purposes the relative positions of these elements are as important as the characteristics of the elements themselves. Although several elements are possible, only a few are needed to describe a face completely. However
these elements do not necessarily correspond to facial features even though they span just a few pixels. Compared to methods such as the PCA, LFA is much more resilient to changes in expression and hence much more robust. It is insensitive to hairstyle changes and growth of facial hair. It is also pose invariant up to 10-15 degrees. However, pose angle can be estimated and compensated for, thereby improving performance. It also works successfully with people wearing eye glasses. It is invariant to gender and race of individuals. Some of the tests in which it was a top performer at FRVT include, high verification accuracy, one-to-many search on a large database, high correct alarm rate for watch list applications, minimal sensitivity to lighting and temporal variation. This state-of-the-art facial recognition system has been deployed in airports, town centers and border crossings worldwide.
CHAPTER 4
DISTANCES IN AFFINE APPROXIMATION SPACES

4.1 Multi-Dimensional Scaling

The affine approximation algorithm utilizes the technique of multi dimensional scaling to find the eigenspace that can approximate the distances computed by any given test face recognition algorithm. There are different types of scaling techniques like classical scaling, metric least squares scaling, uni-dimensional scaling and nonmetric scaling.

Figure 4.1. Find Matrix $A$ such that the Euclidean Distances Between Transformed images are Equal to the Given Distances
4.2 Affine Transformation Based Modelling of Distances

Our goal is to find an affine transformation of the given images so that the Euclidean distance between the transformed images match the given distance set. We build the solution based on the statistical method of multidimensional scaling [15], Let \( n \) faces have dissimilarities \( \{\delta_{rs}\} \) (maybe non-Euclidean) computed between them. The goal of multidimensional scaling (MDS) is to find a configuration of points, representing these faces, in a \( p \) dimensional space such that the distance between two points \( r \) and \( s \) be denoted by \( d_{rs} \) “matches” the corresponding computed dissimilarities. An important distinction of the current work from MDS, is that unlike traditional MDS which just seeks to embed a given set of distances, we also seek a mapping from the input images to MDS coordinates so as to be able to map any new image. But first, a few notational definitions are in order.

Let,

1. \( \vec{x}_i \) be the \( N^2 \times 1 \) sized column vector formed by row scanning the \( N \times N \) \( i \)-th image.  
2. \( \mathbf{X} = [\vec{x}_1, \cdots, \vec{x}_K] \) be the matrix composed out of the image vectors.  
3. \( \delta_{ij} \) be the distance between two images, \( \vec{x}_i \) and \( \vec{x}_j \), that a given algorithm computes. These distances can be arranged as a \( K \times K \) matrix \( \mathbf{D} = [\delta_{ij}^2] \), where \( K \) is the given number of images. Note that the matrix is constructed out of the squared distances.  
4. \( \mathbf{A}(M \times N^2) \) matrix is used to linearly transform the input image vector.  
   \[
   \vec{y}_i = \mathbf{A} \vec{x}_i
   \] (4.1)  
5. The squared Euclidean distance between \( \vec{y}_i \) and \( \vec{y}_j \) is given by,  
   \[
   d_{ij}^2 = (\vec{x}_i - \vec{x}_j)^T (\mathbf{A}^T \mathbf{A}) (\vec{x}_i - \vec{x}_j)
   \] (4.2)  

The distances are stored in a \( K \times K \) matrix \( \mathbf{\Lambda} = [d_{ij}^2] \). 

Let,  
\[
\mathbf{\Lambda} = \mathbf{D}
\] (4.3)
The matrix $A$, which is the affine transform, has to be determined such that

$$(AX)^T(AX) = -\frac{1}{2}H\Lambda H$$  \hspace{1cm} (4.4)$$

where

$$H = (I - \frac{1}{N}\bar{1}\bar{1}^T)$$ \hspace{1cm} (4.5)$$

where $I$ is the identity matrix, $\bar{1}$ is the vector of ones. This operator $H$ is referred to as the centering operator. Applying this operator to both sides of Eq. 4.3, we have

$$(AX)^T(AX) = -\frac{1}{2}HDH = B$$ \hspace{1cm} (4.6)$$

The matrix $B$ is referred to as the “centered” version of $D$. If $D$ is Euclidean then it can be shown that the matrix $B$ is the inner product matrix of the coordinates [15]. We will refer to the transformed coordinates as $X_{MDS}$. Thus,

$$(AX)^T(AX) = (X_{MDS})^T(X_{MDS})$$ \hspace{1cm} (4.7)$$

These coordinates can be arrived at by different MDS embedding schemes, such as classical, least squares, or ISOMAP. However, we chose the simplest possible scheme, the classical scheme [37, 38, 15] that arrives at the solution based on the singular value decomposition of $B = V_{MDS}\Delta_{MDS}V_{MDS}^T$ where $V_{MDS},\Delta_{MDS}$ are the eigenvectors and eigenvalues respectively. Assuming that $B$ represents the inner product distances of an Euclidean distance matrix, the coordinates which are given by

$$X_{MDS} = (V_{MDS}\Delta_{MDS}^{\frac{1}{2}})^T$$ \hspace{1cm} (4.8)$$

This decomposition is possible if the underlying distance matrix $D$ is Euclidean. To handle nonmetric or non-Euclidean dissimilarities and also to handle similarities we first transform them into Euclidean distance. For this, we rely on Gower and Legendre [39, 15], who have shown how dissimilarities can be tested for metric and Euclidean properties and can be transformed to possess these properties if they are absent.
If $D$ is nonmetric then the matrix constructed from elements $\delta_{rc} + c$ (for every $r \neq c$) is metric, where $c \geq \max_{i,j,k} |\delta_{ij} + \delta_{jk} - \delta_{jk}|$.

$D$ is Euclidean if and only if the matrix $B$ is positive semi-definite. If $B$ is positive semi-definite of rank $p$, then a configuration in $p$ dimensional Euclidean space can be found.

If $D$ is a dissimilarity matrix, then there exists a constant $h$ such that the matrix with elements $(\delta_{rs}^2 + h)^{1/2}$ is Euclidean, where $h \geq -2\lambda_n$, the smallest eigenvalue of $B$. In our application context of face recognition, additive constants to the computed dissimilarities do not alter performance.

If $S$ is a positive semi-definite similarity matrix with elements $0 \leq s_{rs} \leq 1$ and $s_{rr} = 1$, then the dissimilarity matrix with elements $\delta'_{rs} = \delta_{rs} + c(1 - \delta_{rs})$ is Euclidean, where $c = -2\lambda_n$ is a constant and $\delta_{rs}$ is Kronecker delta. $\lambda_n$ is the smallest eigenvalue.

To arrive at a solution to Eq. 4.7, we find $A$ such that,

$$X_{MDS} = AX$$

(4.9)

$A$ can be considered to have to two parts, the non rigid part ($A_{nr}$) and the rigid part ($A_r$). Hence, $A$ can also be expressed as $A = A_{nr}A_r$. Eq. 4.9 can now be written as,

$$X_{MDS} = A_{nr}A_rX$$

(4.10)

The rigid part $A_r$ can be arrived at by PCA. Let the PCA coordinates be denoted by $X_{PCA} = A_rX$, where $X_{PCA}$ are the original coordinates projected onto the PCA space. Thus we have,

$$X_{MDS} = A_{nr}X_{PCA}$$

(4.11)

Substituting eq. 4.11 in eq. 4.8 we get,

$$A_{nr}X_{PCA} = (V_{MDS}\Delta_{MDS}^{1/2})^T$$

(4.12)
Now it can be shown that $X_{PCA}X_{PCA}^T = \Lambda_{PCA}$ where $\Lambda_{PCA}$ is the diagonal matrix with the PCA eigenvalues.

$$X_{PCA}X_{PCA}^T = (A_rX)(A_rX)^T$$

$$X_{PCA}X_{PCA}^T = A_r(XX^T)A_r^T$$

$$X_{PCA}X_{PCA}^T = A_r(A_r^T \Lambda_{PCA} A_r)A_r^T$$

However, $A_rA_r^T = I$ where $I$ is an identity matrix. Thus,

$$X_{PCA}X_{PCA}^T = \Lambda_{PCA}$$ (4.13)

Multiplying both sides of eq. 4.12 by $X_{PCA}^T$

$$A_{nr}X_{PCA}X_{PCA}^T = (V_{MDS} \Delta_{MDS}^{1/2})^T X_{PCA}^T$$ (4.14)

Finally, from eq. 4.14 and eq. 4.15 we get,

$$A_{nr} = (V_{MDS} \Delta_{MDS}^{1/2})^T X_{PCA}^T \Lambda_{PCA}^{-1}$$ (4.15)

### 4.2.1 Similarities and Dissimilarities

Face recognition algorithms sometimes compute similarities instead of distances. Similarity coefficients can be converted into dissimilarities or distances using [15].

1. $\delta_{rs} = 1 - s_{rs}$
2. $\delta_{rs} = c - s_{rs}$ for some constant $c$
3. $\delta_{rs} = 2(1 - s_{rs})^{1/2}$

### 4.3 An Alternative Method to Derive $A$

An alternative method to find the affine transformation matrix was proposed by Lakshmi Regina and can be referred to in her masters thesis [19]. It required the computation
of the eigenvectors of a very large matrix and hence was very slow. Our approach is also computationally less expensive. We also achieve the same performance and in the case of ROC better performance with our approach on the same experiments. The approach described by Reguna has been reproduced from her thesis [19] in Appendix A for reference. Some comparison of results are also presented in Appendix A.
CHAPTER 5
EXPERIMENTS AND ANALYSIS OF RESULTS

5.1 Issues Addressed

In this section we present results on a set of studies designed to address the following.

• Can we approximate not only template based algorithms, such as PCA, LDA but also
  feature based algorithms like EBGM and the commercial face recognition algorithm?.

• Does the affine approximation training generalize across data sets collected at differ-
  ent sites?

• Does the training generalize across different covariate affecting face recognition, such
  as, expression and time?

• How close do the recognition performance of the affine approximated algorithm come
  to the original face recognition algorithm?

• What effect, if any, does the distance metric used in the original algorithm, have on
  the performance of Affine Approximation?

5.2 Distance Measures Used by Face Recognition Algorithms

Different algorithms come with different distance measures computed between two im-
ages. Here we summarize the ones that we considered. If an algorithm computes a similarity
measure instead of a distance measure, we discuss how we convert it to a distance measure.

Experiments with the PCA algorithm were performed with different distance metrics
are described below. Let \( \vec{u} \) and \( \vec{v} \) be two images represented as vectors.

City Block(L1):

\[
D_{CityBlock}(\vec{u}, \vec{v}) = \sum_i ||\vec{u}_i - \vec{v}_i||
\]
Euclidean (L2):

\[ D_{Euclidean}(\vec{u}, \vec{v}) = \sqrt{\sum_i (\vec{u}_i - \vec{v}_i)^2} \]

Covariance:

\[ S_{Covariance}(\vec{u}, \vec{v}) = \frac{\sum_i \vec{u}_i \vec{v}_i}{\sqrt{\sum_i \vec{u}_i^2} \sqrt{\sum_i \vec{v}_i^2}} \]

\[ D_{Covariance}(\vec{u}, \vec{v}) = -S_{Covariance}(\vec{u}, \vec{v}) + c \]

where \( c \) is a suitable constant added to convert any negative values to positive ones.

Mahalanobis Space: The Mahalanobis space is the space where the variance along each dimension is one. It can be obtained from the image space by dividing each coefficient of the vector by its corresponding standard deviation. Let \( u \) and \( v \) be vectors in the image space and \( m \) and \( n \) be vectors in the Mahalanobis space. Let \( \Lambda_i \) be PCA eigen values and \( \sigma_i \) be the standard deviation, then \( \Lambda_i = \sigma_i^2 \). The vectors \( u, v \) are related to \( m, n \) in the following manner.

\[ m_i = \frac{u_i}{\sigma_i} \quad n_i = \frac{v_i}{\sigma_i} \]

\( MahaL1 : \)

\[ D_{MahaL1}(u, v) = \sum_i ||m_i - n_i|| \]

\( MahaL2 : \)

\[ D_{MahaL2}(u, v) = \sqrt{\sum_i (m_i - n_i)^2} \]

\( MahaCosine : \)

\[ S_{MahaCosine}(u, v) = \frac{m \cdot n}{||m|| \cdot ||n||} \]

\[ D_{MahaCosine}(u, v) = -S_{MahaCosine} + c \]

where \( c \) is a suitable constant added to convert any negative values to positive ones.

Experiments performed with on the LDA algorithm were performed using the L2 norm.
The Bayesian algorithm has two variants, the Maximum Likelihood and the Maximum a posteriori classifier. For the purposes of this thesis we have used only the Maximum Likelihood classifier.

**ML Similarity Measure:**

\[
S_{ML} = P(\Delta | \Omega_I)
\]

\[
P(\Delta | \Omega_I) = \frac{e^{-0.5 \sum_{i=1}^{T} \frac{y_i^2}{\lambda_i}}}{(2\pi)^{T/2} \prod_{i=1}^{T} \lambda_i^{1/2}}
\]

where \( T \) is the number of truncated dimensions from the original dimensions of the data (in order to reduce computational complexity)

\( \lambda \) are the \( T \) eigen values that span the difference images.

The elastic bunch graphing algorithm provides various features based, as well as geometry based methods to find the similarity between faces. In our experiments we have used a feature based method called the FGNarrowingLocalSearch. This measure is based on average similarity of all the face graph jets based on the \( S_D \) graph Gabor jet similarity.

\[
D = -\log(-S(J, J', \vec{d}))
\]

### 5.3 Data Description

Images from two different data sets, the FERET data set and the Notre Dame data set, were used to test the affine approximation distances. Part of the FERET data set was used for training and part was used for testing. The Notre Dame data set was used for testing. From the FERET data set we used images of type fa (regular facial expressions) and fb (alternate facial expression of the subject taken with the same lighting conditions). They are images of the subjects taken on the same day with the same lighting conditions. Training set as seen in Fig. 5.3 consists of 100 images of 25 subjects with 4 images per subject of both fa, fb type images. It consisted of 2 fa, fb images taken on the same day and 2 fa, fb images taken after a time interval ranging from a few days to a few years. Test set constructed out of the FERET data set as seen in Fig. 5.3 consisted of 600 images of 300 subjects with two images per subject of both fa, fb type images. This test set from
the FERET database was used to conduct experiments involving variation in expression. The PCA, LDA, Bayesian algorithms were trained on this set. There were special training sets for the EBGM and the commercial face recognition algorithms. We used the trained data provided by the CSU implementation because the algorithm required a special tool for ground truthing the images which was not available to us. The EBGM was trained on 68 images of 68 subjects. Similarly, the commercial face recognition algorithm, was trained on the training set that was made available with the software.

The test set from the Notre Dame database as seen in Fig. 5.3 consisted of 830 images from 415 subjects. The images were only of type fa and with similar lighting as the FERET data set. For each subject, two images in the data set with the maximum time difference between them was chosen for each subject. Fig. 5.3 shows a histogram of the time variation in the images. As can be observed from the histogram the variation is concentrated over a period of 100 days. These images were used to conduct experiments involving variation in time. Sample images from the FERET as well as Notre Dame data set can be observed in Fig. 5.3

![Sample Images](image)

Figure 5.1. Training Set: 100 Images with 4 Images of Each Subject of the FERET Data Set

The following preprocessing and normalization code developed at NIST/CSU was used on all the images.

1. Integer to float conversion - Convert 256 gray levels into floating point equivalents.

2. Geometric normalization - Lines up human chosen eye coordinates

3. Masking - Crops the image using an elliptical mask and image borders such that only the face from forehead to chin and cheek to cheek is visible.
4. Histogram equalization - Equalizes the histogram of the unmasked part of the image.

5. Pixel normalization - scales the pixel values to have a mean of zero and a standard deviation of one

5.4 Training and Test Setup

Fig. 5.4 shows the steps of the training phase. First, the face recognition algorithm under consideration is given a set of training images as input and we obtain a distance matrix as output. The baseline PCA algorithm is also given the same set of train images as input. We obtain the projected coordinates of these images in the PCA space and the $A_r$ as output. These PCA coordinates are obtained by retaining all the computed PCA dimensions with non-zero eigenvalues. The Affine Approximation algorithm is given the distance matrix as input along with the projected image coordinates. With these as input, the Affine Approximation algorithm computes the non-rigid part of the affine transfor-
The test setup is shown in Fig. 5.4. The face recognition algorithm under consideration is given a set of test images as input and we get a distance matrix as output. The same set of test images are also projected into the Affine space obtained by the training process.

A \text{nr}. The \text{A}_r from the PCA algorithm and \text{A}_nr from the Affine Approximation algorithm combined form the affine transformation matrix \text{A}.

Figure 5.4. Histogram of the Notre Dame Images Acquired Over Time

Figure 5.5. Sample Images
Figure 5.6. Training Setup for the Affine Approximation algorithm

Figure 5.7. Testing Setup for the Affine Approximation algorithm
We compute Euclidean distances in this projected space which is then compared with the actual distance matrix in terms of biometric performance.

One could compute error measures based on the distances computed, however, since the ultimate goal or task is recognition, we perform evaluation in terms of how well the affine approximated distances can be used in recognition. In our experiments we compare performance using both CMCs and ROCs.

5.5 Affine Matrix

It is worthwhile to visualize the affine transformation matrix. Figs. 5.5, 5.5, 5.5, 5.5 show the $A_{nr}^T A_{nr}$ for each of the algorithms where $A_{nr}$ represents the non rigid part of the Affine transformation matrix. $A_{nr}^T A_{nr} = I$ if the PCA space does not need to be modified. From the varying values along the diagonal we can see that in BIC and EBGM algorithms are well approximated by the PCA dimension with shears and stretch along these dimensions. However, the significant non-zero off-diagonal values for the LDA and the commercial algorithm denotes that we need to shear and stretch the PCA space along dimensions that are not aligned along the PCA dimensions. Fig. 5.5 shows the plot of the diagonal values of the $A_{nr}^T A_{nr}$ for each algorithm versus $\frac{1}{\sqrt{\lambda_i^{PCA}}}$. We note that there is steady increase in the amount of stretch or shear, in the affine transformation matrix, along the dimensions with lower PCA eigenvalues. This implies that the least dominant PCA dimensions are undergoing the maximum amount of transformation.

5.5.1 Affine Space Dimensions

We can also look at top three affine space dimensions, visualized as faces, for all the face recognition algorithms. The dimensions capture the set of dominant features used to characterize a face. Each of these images highlight the variation in a particular feature for the given set of images. Both very dark and very bright region signify importance. The top dimensions with the maximum eigen value will thus signify features with maximum variation across subjects. In Fig. 5.5.1 we take a look at the PCA eigenvectors before and after they have been transformed by the affine transformation. We see that the most
important feature to the PCA is the lighting as can be observed from the intense contrast of bright and dark regions along the two halves of the face. Other template matching algorithms like the LDA and the BIC also capture similar features as the principal component of variation. However, the EBGM and the commercial face recognition algorithm, which
are feature based algorithms, use more local facial features as in captured in the isolated bright and dark patches in the eigen-dimensions.

Fig 5.5.1 show the PCA eigenfaces along which there is maximum need for shear and stretch by the Affine Approximation algorithm. We see that local features are being re-emphasized.
Figure 5.12. Visualization of Diagonal Values of $A_{nr}^T A_{nr}$ for All Algorithms
Figure 5.13. Top Dimensions of the Affine Approximation to the Different Algorithms. Last Row Shows the Corresponding PCA Dimension for Comparison
Figure 5.14. PCA Eigen Dimensions Along Which We Need to Stretch and Shear the Most to Match Distances
5.6 Performance on Data with Time Variation

In this section, we look at the results of experiments with all the different algorithms performed on the Notre Dame data set. In these experiments, the Affine Approximation has been trained on the 25 subjects in FERET data set. The gallery and probe set consisted of 415 images each of 415 subjects in the Notre Dame data set. The gallery set contained the images of these 415 subjects when they first taken. The probe set containing the images of the same 415 subjects with the maximum time gap available from the subsequent times when the images were re-acquired.

5.6.1 Identification and Verification Performance

Figure 5.15. ROC on Notre Dame Images Showing the Performance Comparison of the Different Face Recognition Algorithm along with the Affine Approximation Algorithm (a) LDA (b) BIC (c) EBGM (d) Commercial
The verification performance is shown as ROC curves in Fig. 5.6.1 for both the Affine Approximation algorithm and the face recognition algorithm. The identification performance are shown in Fig. 5.6.1 We note that,

- The CMC and ROC performance of all the algorithms are well approximated.
- In case of BIC and EBGM algorithms, the recognition performance is higher than the original algorithms.

![CMC Curve](a) ![CMC Curve](b) ![CMC Curve](c) ![CMC Curve](d)

Figure 5.16. CMC on Notre Dame Images Showing the Performance Comparison of the Different Face Recognition Algorithm along with the Affine Approximation Algorithm (a) LDA (b) BIC (c) EBGM (d) Commercial
5.7 Performance on Data With Expression Variation

In this section, we present the results of testing on the data that involve expression change. As before the train set consists of 100 images of 25 subjects with 4 images per subject. The test set consists of 600 images of 300 subjects. The gallery contained of 300 "fa" images and the probe consisted of 300 "fb" images in the FERET data set. There is no overlap between the identity of subjects in the train set and the test sets.

5.7.1 Identification and Verification Performance

![ROC curve](image)

Figure 5.17. ROC on FERET Data Set Showing the Performance Comparison of the Different Face Recognition Algorithm along with the Affine Approximation Algorithm. (a) LDA (b) BIC (c) EBGM (d) Commercial

From the verification performance presented in Fig. 5.7.1 we see that the Affine Approximation algorithm performs as well as the LDA algorithm and in case of BIC, has a
better performance. In case of the EBGM, when we compare the performance in graph (c) of Fig. 5.6.1 and graph (c) in Fig. 5.7.1 algorithm we notice there is a drop in the performance. This is probably due to the fact that the EBGM algorithm and its Affine Approximation was different. EBGM was trained on just "fa" images and the Affine Approximation algorithm training included "fa" and "fb" images. For the Notre Dame data set, which consists of only "fa" images(variation in time), we get a better performance approximation. We also observe from graph (d) in Fig. 5.7.1 that the performance of Commercial face recognition algorithm is better at low false alarm, but the performance of the Affine Approximation algorithm is much better at higher values of false alarm. The identification performance which is presented in Fig. 5.7.1 also follows a similar pattern.

Figure 5.18. CMC on FERET Data Set Showing the Performance Comparison of the Different Face Recognition Algorithm along with the Affine Approximation Algorithm (a) LDA (b) BIC (c) EBGM (d) Commercial
5.8 Approximating PCA with Different Distance Measures

![ROC curves](image)

**Figure 5.19. ROC Curves of PCA Algorithm with Different Distance Measures**

(a) Euclidean (b) Covariance (c) Maha-Cosine (d) MahL1

Fig. 5.8, 5.8 show the CMC and ROC curves respectively when the PCA algorithm is used as the test face recognition algorithm. For the PCA, the non-rigid part, $A_{nr}$, should be an identity matrix. We show results on the 600 images from 300 subjects from the FERET data set. As before, the gallery consists of all "fa" images and the probe consists of all "fb" images. We observe from graph (a) in Fig. 5.8 and graph (a) in 5.8 that both the ROC and the CMC curves validate this fact. In Fig. 5.8 we observe that the Affine Transformation matrix $A_{nr}$, visualized as $A_{nr}^T A_{nr}$, is an identity matrix.

From the graphs in Fig. 5.8, 5.8 we can also observe the effects on the Affine Approximation algorithm of using different distance along with the PCA. We observe from
graphs (c) and (d) in Fig. 5.8 and Fig. 5.8 that PCA with Mahanalobis cosine distance and Mahanalobis L1 distance also are well approximated. The performance of the affine approximation is better than that of the PCA+Covariance distance measure as seen in Fig. 5.8(b) and Fig. 5.8(b).

5.9 Effects of Normalization on the Distance Matrix

From Fig. 5.7.1 and 5.7.1 we observe that there is a significant drop in the performance of the Commercial algorithm as compared to the other test algorithms at low false alarm rates. Although the reason for this is not clear at present and more research needs to be done to investigate the reasons for the gap in the performance, one of the factors affecting the approximation could be the unknown normalization performed on the distance
Figure 5.21. Visualization of $A_{nr}^T A_{nr}$ for PCA Algorithm (Euclidean Distance) for the FERET Data Set

Figure 5.22. Visualization of $A_{nr}^T A_{nr}$ for PCA Algorithm (Covariance Distance) for the FERET Data Set
Figure 5.23. Visualization of $A^T_{nr}A_{nr}$ for PCA Algorithm (MahCosine Distance) for the FERET Data Set

Figure 5.24. Visualization of $A^T_{nr}A_{nr}$ for PCA Algorithm (MahL1 Distance) for the FERET Data Set
matrix output by the Commercial Face recognition algorithm. In Fig. 5.9 we see the effect of different normalization procedures on the distance scores output by the Affine Approximation algorithm on the ROC curves. We observe that after the Z normalization, there is a definite shift in the ROC curve towards the original ROC curve output by the face recognition algorithm.

- G - the images in the gallery.
- P - the images in the probe.
- $S_{pg}$ - match score of a probe 'p' with gallery template 'g'.
- $S_{pG}$ - vector of match scores obtained when a probe 'p' with entire gallery G.

In min-max normalization, the minimum and maximum scores are used to normalize the score vector according to equation 5.1

$$S'_{pG} = \frac{S_{pG} - \text{min}(S_{pG})}{\text{max}(S_{pG} - \text{min}(S_{pG}))}$$ (5.1)

In Z-normalization, using the mean and standard deviation of the score vector $S_{pG}$, we get the normalized score as in equation 5.2. The resulting $S'_{pG}$ have a mean of zero and a standard deviation of one.

$$S'_{pG} = \frac{S_{pG} - \text{mean}(S_{pG})}{\text{std}(S_{pG})}$$ (5.2)
Figure 5.25. Effect of Normalization on the ROC Curves for the Commercial Face Recognition Algorithm on the FERET Data Set
CHAPTER 6
SUMMARY AND CONCLUSION

In this thesis, we have proposed an approach to model distance functions of face recognition algorithms so as to be able to evaluate the algorithms at a deeper level. Given any distance matrix computed by a face recognition algorithm, we "learn" an affine transformation of the PCA eigen space such that it can match the distances. We have tested this affine transformation on two different sets of test data, the FERET database having face images with variation in expression and time and Notre Dame database having images with variation in time. We have also performed our experiments on both template matching algorithms like the BIC and LDA as well as graph based algorithms like EBGM and the commercial face recognition algorithm.

The core technique used to compute the affine transformation matrix is classical multidimensional scaling. Multidimensional scaling allows data with a high number of dimensions to be embedded in a lower dimensional space. To augment this process, we also "learn" a mapping function so as to be able to compute distance between any two images. This is unlike a plain MDS that just provides an embedding of the given data.

We have used two different databases with two different covariates such as expression and time to test the performance of the affine approximation. Since the approximation is quite close both sets of test data, we can conclude that the affine approximation training can be generalized across covariates.

Since the ultimate goal, is face recognition it is important to also discuss how the recognition performance of the affine approximation algorithm and the original face recognition compare. As we have seen in the previous chapter, from the ROCs and CMCs, the recognition performance of the affine approximation algorithm is quite close to the original face recognition algorithm and in some cases like the BIC, even performs better. We find
that the Affine Approximation algorithm approximates the face recognition algorithms in
the following rank order. LDA is best approximated followed by BIC, commercial, and
EBGM for the data with expression variation. For the data with time variation we find
that the order of the rank is BIC, LDA, EBGM, commercial. From the experiments of
the PCA with different distance measures, we find that PCA+Euclidean distance is best
approximated followed by PCA+MahL1, PCA+MahCosine, and PCA+Covariance.

The affine approximation algorithm, can serve as an important tool to evaluate other
face recognition algorithm. For face recognition algorithms to mature into more efficient
systems, it is not only important to develop good baseline algorithms but also essential to
develop good evaluation protocols.

6.1 Future

More research and investigation is needed to see how well the affine transformations
perform with data sets with covariates which have not been explored in this thesis. We also
need to study the effect of distance measures and normalization techniques on the quality
of affine transformations.
REFERENCES


APPENDICES
Appendix A Comparison of Results with Reguna

The following notation will be used to describe the linear transform strategy used by Reguna.

1. Let \( \vec{x}_i \) be the \( N^2 \times 1 \) sized column vector formed by row scanning the \( N \times N \) \( i \)-th image.

2. Let \( K \) denote the number of images.

3. Let \( d^A_{ij} \) be the distance between \( \vec{x}_i \) and \( \vec{x}_j \) that a given algorithm computes. These distances can be arranged as a \( K \times K \) matrix \( D \), where \( K \) is the given number of images.

4. Let the matrix \( A \) be a \( M \times N^2 \) sized array that is used to linearly transform the input image vector.

\[
\vec{y}_i = A \vec{x}_i \quad (A.1)
\]

The rows of the matrix \( A \) denote the axes of the reduced \( M \) dimensional space. For a PCA based space, the rows of \( A \) will be orthogonal to each other.

5. The (square) Euclidean distance between \( \vec{y}_i \) and \( \vec{y}_j \) can be denoted by

\[
d^E(\vec{y}_i, \vec{y}_j) = \sum_{k=1}^{M} (\vec{y}_i(k) - \vec{y}_j(k))^2
= (\vec{y}_i - \vec{y}_j)^T (\vec{y}_i - \vec{y}_j) \quad (A.2)
\]

Problem Definition : The matrix \( A \), which is the affine transform, has to be determined such that

\[
d^E(\vec{y}_i, \vec{y}_j) = d^A_{ij} \quad (A.3)
\]

\[
d^E(\vec{y}_i, \vec{y}_j) = (\vec{y}_i - \vec{y}_j)^T (\vec{y}_i - \vec{y}_j)
= (A\vec{x}_i - A\vec{x}_j)^T (A\vec{x}_i - A\vec{x}_j)
= (A(\vec{x}_i - \vec{x}_j))^T (A(\vec{x}_i - \vec{x}_j))
= (\vec{x}_i - \vec{x}_j)^T (A^T A)(\vec{x}_i - \vec{x}_j) \quad (A.4)
\]
Appendix A (Continued)

(As an aside, it is worth noting that if the rows of A were orthonormal (e.g. in PCA) and size of A was $N^2 \times N^2$ then $A^T A = A A^T = I$, the identity matrix. Or in other words, Euclidean distance are preserved: $d^E(\vec{y}_i, \vec{y}_j) = d^E(\vec{x}_i, \vec{x}_j)$).

Let,

1. $B = A^T A$, where $B$ is a $N^2 \times N^2$ sized matrix. Note that $B$ is symmetric, i.e. $B^T = B$.

2. $\vec{e}_{ij} = \vec{x}_i - \vec{x}_j$ is a $N^2 \times 1$ sized column vector.

Using the above notations

\[
d^E(\vec{y}_i, \vec{y}_j) = \vec{e}_{ij}^T B \vec{e}_{ij} = \sum_{k=1}^{N^2} \sum_{l=1}^{N^2} B(k, l) \vec{e}_{ij}(k) \vec{e}_{ij}(l) \tag{A.5}
\]

The above double sum can be expressed as product of two column vectors as follows. Let two columns vectors be defined as follows

1. $\vec{b}$ is a $\frac{N^2(N^2+1)}{2}$ sized column vector by scanning the lower triangular entries (including the diagonal) of B. Thus,

\[
\vec{b}(\frac{k(k+1)}{2} + l) = B(k, l), \text{ for } l \leq k, k = 1, \cdots, N^2 \tag{A.6}
\]

2. $\vec{\psi}_{ij}$ is a $\frac{N^2(N^2+1)}{2}$ sized column vector such that

\[
\vec{\psi}_{ij}(\frac{k(k+1)}{2} + l) = \begin{cases} 
\vec{e}_{ij}(k)^2 & \text{for } k = l \\
2\vec{e}_{ij}(k)\vec{e}_{ij}(l) & \text{for } l \leq k 
\end{cases} \tag{A.7}
\]

Using the above equation

\[
d^E(\vec{y}_i, \vec{y}_j) = \vec{\psi}_{ij}^T \vec{b} \tag{A.8}
\]

$\vec{b}$ should be determined such that

\[
\vec{\psi}_{ij}^T \vec{b} = d^A_{ij} \tag{A.9}
\]
for every pair of images. These $\frac{K(K-1)}{2}$ equations can be compactly expressed in matrix notion as follows

$$\psi^T \vec{b} = \vec{d}^A$$

(A.10)

where $\psi$ is a $\frac{K(K-1)}{2} \times \frac{N^2(N^2+1)}{2}$ sized matrix formed by concatenating the column vectors $\tilde{\psi}_{ij}$. And, $\vec{d}^A$ is column vector of the given distances.

In the above equation, $\vec{b}$ is unknown and can be solved using any standard linear equation solver. The only constraint is that $K \geq N^2 + 1$ so that the number equation is at least equal to the number of unknowns. Given $\vec{b}$, the matrix B can be formed, from which we would like to form A such that $B = A^TA$. This can be done using the eigenvectors ($\vec{u}_i$) and eigenvalues ($\lambda_i$) of $B$.

The matrix $B$ is factored into $U\Lambda U^T$, where the columns of $U$ are the eigenvectors, $\vec{u}_i$, of $B$ and $\Lambda$ is a diagonal matrix formed out of the eigenvalues. Since $B$ is guaranteed by the fact that $B$ is symmetric. In fact, we can also claim that the eigenvalues would real and positive. Symmetric matrices have real eigenvalues. From Eq. A.5 it follows that $B$ is positive semi-definite because distances are always are greater than or equal to zero. And, eigenvalues of positive semi-definite matrices are greater than or equal to zero.

Given the eigenvalue and eigenvector decomposition of $B$ we can choose

$$A = \Lambda^{\frac{1}{2}} U^T$$

(A.11)

or in other words the rows of $A$ are the scaled eigenvectors $\vec{u}_i$’s. In particular, the $i$-th row of $A$ will be $\sqrt{\lambda_i} \vec{u}_i^T$. Thus, the non-zero rows of $A$ would be determined by the number of non-zero eigenvalues of $B$. $A$ is the affine approximation matrix which when will attempt to duplicate the results of any input face recognition algorithm.
Appendix A (Continued)

Here we have shown a simultaneous comparison of some of the results got by us and those by Regina using her approach on the same experiments. These experiments were conducted on the FERET data base using a training set of 100 images with 4 images per subject. The test set consisted of 600 images of 300 subjects. The gallery set contained 300 images of type "fa" and the probe set consisted of 300 images of type "fb".

We can observe from Fig. A that the CMCs from our approach and her approach is almost identical. However the ROCs show a remarkable improvement by our method as can be observed in Fig. A.
Appendix A (Continued)

![CMC Curves for Different Algorithms Using FERET Data](image.png)

Figure A.1. Comparison of CMC Using FERET Data on Different Algorithms
Appendix A (Continued)

Figure A.2. Comparison of ROC Using FERET Data on Different Algorithms