An indepth analysis of face recognition algorithms using affine approximations.

Lakshmi Reguna
University of South Florida

Follow this and additional works at: http://scholarcommons.usf.edu/etd
Part of the American Studies Commons

Scholar Commons Citation
http://scholarcommons.usf.edu/etd/1457
AN INDEPTH ANALYSIS OF FACE RECOGNITION ALGORITHMS USING AFFINE APPROXIMATIONS

by

LAKSHMI REGUNA

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

Major Professor: Sudeep Sarkar, Ph.D.
Dmitry Goldgof, Ph.D.
Nagarajan Ranganathan, Ph.D.

Date of Approval:
May 19, 2003

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

© Copyright 2003, Lakshmi Reguna
DEDICATION

To my husband and my parents without whom I would not have been able to come so far
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. Goldgof and Dr. Ranganathan for being a part of my thesis committee. I also would like to Dr. P. Jonathon Phillips from NIST for his valuable inputs. I would also like to thank Dr. Ross Beveridge and his students from the Colorado State University for their patience and help.
# TABLE OF CONTENTS

LIST OF TABLES  ii  
LIST OF FIGURES  iii  
ABSTRACT  vi  

CHAPTER 1  RELATED WORK  
1.1 The FERET Protocol  
1.1.1 FERET 1994-1996  3  
1.1.2 Facial Recognition Vendor Test  5  
1.2 Evaluating Statistical Significance of the Results  
1.2.1 Significance of the Affine Approximating Algorithm  8  

CHAPTER 2  AFFINE APPROXIMATION ALGORITHM  9  

CHAPTER 3  TEST ALGORITHMS  
3.1 Principal Component Analysis (PCA) Algorithm  15  
3.2 Linear Discriminant Analysis (LDA) Algorithm  16  
3.3 Bayesian Intrapersonal and Extrapersonal Classifier  17  

CHAPTER 4  EXPERIMENTAL SETUP  
4.1 Data Description  18  
4.2 Training Scenarios  18  

CHAPTER 5  ANALYSIS OF RESULTS  
5.1 Visualization of Distance Matrixes  21  
5.2 Gross Error Measures  22  
5.3 Analysis of Affine Approximation Algorithm  30  
5.4 Eigen Values of B Matrix  35  
5.5 Performance of Identification and Verification Scenarios  41  
5.6 McNemar’s Test  41  
5.7 Difference Caused by Training Scenarios  52  

CHAPTER 6  CONCLUSION  
6.1 Future Work  54  

REFERENCES  55  

APPENDICES  58  
Appendix A  More Results  59


LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>McNemar’s Test for Training Scenario 1 at Rank 1.</td>
<td>51</td>
</tr>
<tr>
<td>5.2</td>
<td>McNemar’s Test for Training Scenario 2 at Rank 1.</td>
<td>51</td>
</tr>
<tr>
<td>5.3</td>
<td>McNemar’s Test for Training Scenario 2.</td>
<td>51</td>
</tr>
<tr>
<td>A.1</td>
<td>Training Scenario 2: Average RMS Error.</td>
<td>61</td>
</tr>
<tr>
<td>A.2</td>
<td>Training Scenario 2: Stress 1.</td>
<td>61</td>
</tr>
<tr>
<td>A.3</td>
<td>McNemar’s Test for Training Scenario 1 at Rank 1.</td>
<td>74</td>
</tr>
<tr>
<td>A.4</td>
<td>McNemar’s Test for Training Scenario 2 at Rank 1.</td>
<td>74</td>
</tr>
<tr>
<td>A.5</td>
<td>McNemar’s Test for Training Scenario 2.</td>
<td>74</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.1</td>
<td>Problem Definition.</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Flow Chart of the Affine Approximation Algorithm.</td>
<td>13</td>
</tr>
<tr>
<td>4.1</td>
<td>Training Scenario 1: Block Diagram.</td>
<td>19</td>
</tr>
<tr>
<td>4.2</td>
<td>Training Scenario 2: Block Diagram.</td>
<td>19</td>
</tr>
<tr>
<td>5.1</td>
<td>Visualization of Similarity Matrix for the PCA Algorithm Using the Euclidean Distance Measure.</td>
<td>23</td>
</tr>
<tr>
<td>5.2</td>
<td>Visualization of Similarity Matrix for the PCA Algorithm Using the Cosine Distance Measure.</td>
<td>24</td>
</tr>
<tr>
<td>5.3</td>
<td>Visualization of Similarity Matrix for the LDA Algorithm.</td>
<td>25</td>
</tr>
<tr>
<td>5.4</td>
<td>Visualization of Similarity Matrix for the Bayesian Algorithm.</td>
<td>27</td>
</tr>
<tr>
<td>5.5</td>
<td>Training Scenario 1: Plot of RMS Error.</td>
<td>28</td>
</tr>
<tr>
<td>5.6</td>
<td>Training Scenario 1: Stress Plot.</td>
<td>28</td>
</tr>
<tr>
<td>5.7</td>
<td>Training Scenario 2: Plot of RMS Error.</td>
<td>29</td>
</tr>
<tr>
<td>5.8</td>
<td>Training Scenario 2: Stress Plot.</td>
<td>29</td>
</tr>
<tr>
<td>5.9</td>
<td>Training Scenario 1: Visualization of Affine Transformation Matrix for the PCA Algorithm Using the Euclidean Distance Measure.</td>
<td>31</td>
</tr>
<tr>
<td>5.10</td>
<td>Training Scenario 1: Visualization of Affine Transformation Matrix for the PCA Algorithm Using the Cosine Distance Measure.</td>
<td>31</td>
</tr>
<tr>
<td>5.11</td>
<td>Training Scenario 1: Visualization of Affine Transformation Matrix for the LDA Algorithm.</td>
<td>32</td>
</tr>
<tr>
<td>5.12</td>
<td>Training Scenario 1: Visualization of Affine Transformation Matrix for the Bayesian Algorithm.</td>
<td>32</td>
</tr>
<tr>
<td>5.13</td>
<td>Training Scenario 2: Visualization of Affine Transformation Matrix for the PCA algorithm Using the Euclidean Distance Measure.</td>
<td>33</td>
</tr>
<tr>
<td>5.14</td>
<td>Training Scenario 2: Visualization of Affine Transformation Matrix for the PCA Algorithm Using the Cosine Distance Measure.</td>
<td>33</td>
</tr>
</tbody>
</table>
Figure 5.15 Training Scenario 2: Visualization of Affine Transformation Matrix for LDA Algorithm.

Figure 5.16 Training Scenario 2: Visualization of Affine Transformation Matrix for Bayesian Algorithm.

Figure 5.17 Training Scenario 1: Eigen Values for the PCA Algorithm Using the Euclidean Distance Measure.

Figure 5.18 Training Scenario 1: Eigen Values for the PCA Algorithm Using the Cosine Distance Measure.

Figure 5.19 Training Scenario 1: Eigen Values for the LDA Algorithm.

Figure 5.20 Training Scenario 1: Eigen Values for the Bayesian Algorithm.

Figure 5.21 Training Scenario 2: Eigen Values for the PCA Algorithm Using the Euclidean Distance Measure.

Figure 5.22 Training Scenario 2: Eigen Values for the PCA Algorithm Using Cosine Distance Measure.

Figure 5.23 Training Scenario 2: Eigen Values for the LDA Algorithm.

Figure 5.24 Training Scenario 2: Eigen Values for the Bayesian Algorithm.

Figure 5.25 Training Scenario 1: CMC Curve for PCA Algorithm Using the Euclidean Distance Measure.

Figure 5.26 Training Scenario 1: ROC Curve for PCA Algorithm Using the Euclidean Distance Measure.

Figure 5.27 Training Scenario 1: CMC Curve for PCA Algorithm Using Cosine Distance Measure.

Figure 5.28 Training Scenario 1: ROC Curve for PCA Algorithm Using Cosine Distance Measure.

Figure 5.29 Training Scenario 1: CMC Curve for LDA Algorithm.

Figure 5.30 Training Scenario 1: ROC Curve for LDA Algorithm.

Figure 5.31 Training Scenario 1: CMC Curve for Bayesian Algorithm.

Figure 5.32 Training Scenario 1: ROC curve for Bayesian Algorithm.

Figure 5.33 Training Scenario 2: CMC Curve for PCA Algorithm Using the Euclidean Distance Measure.

Figure 5.34 Training Scenario 2: ROC Curve for PCA Algorithm Using the Euclidean Distance Measure.
Figure 5.35 Training Scenario 2: CMC Curve for PCA Algorithm Using Cosine Distance Measure.

Figure 5.36 Training Scenario 2: ROC Curve for PCA algorithm Using Cosine Distance Measure.

Figure 5.37 Training Scenario 2: CMC Curve for LDA Algorithm.

Figure 5.38 Training Scenario 2: ROC Curve for LDA Algorithm.

Figure 5.39 Training Scenario 2: CMC Curve for Bayesian Algorithm.

Figure 5.40 Training Scenario 2: ROC Curve for Bayesian Algorithm.

Figure 5.41 Eigen Faces of Set $C_1$.

Figure 5.42 Eigen Faces of Set $C_2$.

Figure A.1 Eigen Faces of Set $C_1$.

Figure A.2 Eigen Faces of Set $C_2$.

Figure A.3 Training Scenario 1 - Plot of RMS Error.

Figure A.4 Training Scenario 1 - Stress Plot.

Figure A.5 Training Scenario 1: CMC Curve for PCA Algorithm.

Figure A.6 Training Scenario 1: ROC Curve for PCA Algorithm.

Figure A.7 Training Scenario 1: CMC Curve for LDA Algorithm.

Figure A.8 Training Scenario 1: ROC Curve for LDA Algorithm.

Figure A.9 Training Scenario 1: CMC Curve for Bayesian Algorithm.

Figure A.10 Training Scenario 1: ROC Curve for Bayesian Algorithm.

Figure A.11 Training Scenario 2: CMC Curve for PCA Algorithm.

Figure A.12 Training Scenario 2: ROC Curve for PCA Algorithm.

Figure A.13 Training Scenario 2: CMC Curve for LDA Algorithm.

Figure A.14 Training Scenario 2: ROC Curve for LDA Algorithm.

Figure A.15 Training Scenario 2: CMC Curve for Bayesian Algorithm.

Figure A.16 Training Scenario 2: ROC Curve for Bayesian Algorithm.
AN INDEPTH ANALYSIS OF FACE RECOGNITION ALGORITHMS USING AFFINE APPROXIMATIONS

Lakshmi Reguna

ABSTRACT

In order to foster the maturity of face recognition analysis as a science, a well implemented baseline algorithm and good performance metrics are highly essential to benchmark progress. In the past, face recognition algorithms based on Principal Components Analysis (PCA) have often been used as a baseline algorithm. The objective of this thesis is to develop a strategy to estimate the best affine transformation, which when applied to the eigen space of the PCA face recognition algorithm can approximate the results of any given face recognition algorithm. The affine approximation strategy outputs an optimal affine transform that approximates the similarity matrix of the distances between a given set of faces generated by any given face recognition algorithm. The affine approximation strategy would help in comparing how close a face recognition algorithm is to the PCA based face recognition algorithm. This thesis work shows how the affine approximation algorithm can be used as a valuable tool to evaluate face recognition algorithms at a deep level.

Two test algorithms were chosen to demonstrate the usefulness of the affine approximation strategy. They are the Linear Discriminant Analysis (LDA) based face recognition algorithm and the Bayesian interpersonal and intrapersonal classifier based face recognition algorithm. Our studies indicate that both the algorithms can be approximated well. These conclusions were arrived based on the results produced by analyzing the raw similarity scores and by studying the identification and verification performance of the algorithms. Two training scenarios were considered, one in which both the face recognition and the affine approximation algorithm were trained on the same data set and in the other, different data sets were used to train both the algorithms. Gross error measures like the average RMS error and Stress-1 error were used to directly compare the raw similarity
scores. The histogram of the difference between the similarity matrixes also clearly showed that the error spread is small for the affine approximation algorithm. The performance of the algorithms in the identification and the verification scenario were characterized using traditional CMS and ROC curves. The McNemar’s test showed that the difference between the CMS and the ROC curves generated by the test face recognition algorithms and the affine approximation strategy is not statistically significant. The results were statistically insignificant at rank 1 for the first training scenario but for the second training scenario they became insignificant only at higher ranks. This difference in performance can be attributed to the different training sets used in the second training scenario. We believe that this difference in performance between the first and the second training scenario can be reduced by using a larger training set.
CHAPTER 1
RELATED WORK

The rapid development of face recognition technology necessitated the development of effective protocols and techniques to evaluate the performance of face recognition algorithms. If automatic face recognition systems have to be deployed in real world scenarios then effective means of comparing the performances of independent face recognition systems have to be developed. Researchers used the insight gained from publicly available protocols to evaluate biometric systems for developing methods to evaluate face recognition system.

The UK Biometrics Working Group’s Biometric Test Programme Report compared six different biometrics [12]. The report is the first evaluation that directly compares performance of different biometrics for the same application. Face, fingerprint, hand geometry, iris, vein and voice recognition systems were tested for verification in a normal office environment with cooperative, non-habituated users. The NIST speaker recognition evaluations measure verification performance [14]. The protocols used for evaluating fingerprint technology also greatly helped in developing methods to evaluate face recognition systems [7, 8, 9]. The Biometric Testing Center at San Jose State University explored a number of essential questions relating to the science underpinning biometric technologies. The results of their endeavors contains evaluation results from the INSPASS Hand Geometry System, the Philippine AFIS System and numerous other small scale evaluations [13].

The most important lesson learnt from these evaluations is that large sets of test images are essential for adequate evaluation. Also the sample should be statistically as similar as possible to the images that would arise in the application considered. The costs of errors in recognition should be reflected in the scoring process. The reject error behavior should be studied and not just forced recognition. The operation of a pattern recognition
systems is statistical and distributions of success and failure are simply non-existent. The
distributions of success and failure depend heavily on the application being considered. Also
no theory exists that can predict these distributions for new applications. So the evaluation
protocol should be designed such that it is based on the application being considered as
closely as possible.

Philips et al [20, 12] identified three basic scenarios in which biometric systems should
be evaluated. They are as follows:


2. Scenario Evaluation.


Technology evaluation attempts to compare competing algorithms from a single tech-
nology. The testing of all the algorithms is carried out on a standardized database. The
data would be collected by a universal sensor. So a database should be created such
that it is neither too difficult nor too easy for the algorithms being tested. Although sample
data may be distributed for developmental or tuning purposes, the actual testing would
be done on data not previously seen by algorithm developers. Testing is carried out using
offline processing of the data. Since the database is fixed, the results are repeatable. Tech-
nological evaluations have been very crucial in understanding the strengths and weakness
of biometric systems.

The goal of scenario evaluation is to determine the overall system performance in a
prototype or simulated application. Testing is carried out on a complete system in an
environment that attempts to model a real-world target application of interest. Each
tested system would have its own acquisition sensor and so will receive slightly different
data. However care should be taken such that the data collection across all tested systems
is in the same environment with the same population. The testing maybe a combination
of offline and online comparisons. The extent to which the test results would be repeatable
would directly depend on how controlled the testing environment is maintained.

Operational evaluation aims to determine the performance of a complete biometric sys-
tem in a specific application environment with a specific target population. Offline testing
maybe possible depending upon the data capabilities of the tested device. Operational test results will not be repeatable because of unknown and undocumented differences between operational environments.

The evaluation protocol determines how a biometric system is tested, test data is selected and measures of performance are chosen. It should neither be easy nor too hard. Also the evaluation protocol should spread the range of performance scores over a range so that it is easy to distinguish between the different algorithms. Also biometric systems should be tested on previously unseen data. If the biometric systems are not tested on previously unseen data then it would only help in measuring how a biometric system tunes to a particular data set [20]. One of the most significant evaluation protocols in face recognition is the FERET evaluation protocol [16, 17, 18]. The subsequent section deals with the FERET protocol in more detail.

1.1 The FERET Protocol

The most important step in the direction of developing a standardized technique for evaluating face recognition systems was brought about by the FERET program. Before the FERET program was developed most of the face recognition systems reported perfect performance in a small data set of images. But when deployed in a real world scenario the reality was far from the reported performance.

1.1.1 FERET 1994-1996

Three FERET evaluations for laboratory algorithms was carried out in 1994, 1995 and 1996 [16, 17, 19]. This was followed by evaluation of commercial face recognition systems in 2000 and 2002 [24, 23]. As a part of the FERET program a large database of still images was collected between Aug 1993 and July 1996 that consists of 14,126 images of 1199 individuals. Later the FERET Vendor Test [23] conducted in 2002 used a much larger database consisting of 121,589 operational images of 37,437 subjects. The images were provided from the U.S Department of State’s Mexican non-immigrant visa archive.
The FERET evaluation protocol designed for the tests during 1994, 1995 and 1996 was a general evaluation strategy to measure the performance of laboratory algorithms on a common database. The FERET test was not concerned with the performance of individual components of an algorithm nor was it concerned with the performance of the algorithms under various operational scenarios. The algorithms were evaluated against different categories of images in order to obtain a robust assessment of performance. The images differed by changes in the illumination and lighting, presence or absence of glasses and the time of acquisition of images of the same subject.

The first FERET evaluation test was administered in August 1994 [19]. During this evaluation the PCA face recognition algorithm was used as a baseline algorithm to compare the performance of other face recognition algorithms. This evaluation could measure the performance of algorithms that could automatically locate, normalize and identify faces. The FERET protocol defined the gallery set as the set of known individuals and the probe set as the set of unknown individuals [16]. The evaluation in August 1994 consisted of three tests, each with a different probe and gallery set. The first test measured the identification performance from a gallery of 316 individuals with one image per person. The second test was a false alarm test which measured how well an algorithm rejects faces not in the gallery. The third test baselined the effects of pose changes on performance.

The FERET protocol characterized the performance of the face recognition algorithms in the identification scenario using Cumulative Match Score (CMS) curves and the performance in the verification scenario using Receiver-Operator Characteristics (ROC) curves. The CMS curve plots the percentage of queries in which the correct answer can be obtained within a certain rank. The ROC curves plots the false reject error versus the false alarm error.

Mansfield and Wayman [12] also recommended the use of Detection Error Trade-off (DET) curves to report the performance of biometric systems. DET curve plot the false match rate versus the false non-match rate. They defined the false match rate (FMR) as the expected probability that a sample will be falsely declared to match a single randomly selected "non-self" template" and the false non-match rate (FNMR) as "the expected probability that a sample will be falsely declared not to match a template of the same
measure from the same user applying the sample”. The DET curve is essentially the same as the ROC curve; it plots the two error rates against each other instead of the detection versus false alarm rate as in the ROC curve. False reject is 1 - detection rate. False alarm and false reject rates are computed over the number of comparisons whereas the false match/non-match rate are computed over the number of transactions made by the user.

The second FERET test was administered in March 1995. One of the main emphasis of this test was on duplicate probes. A duplicate probe is usually an image of a person whose corresponding gallery image was taken on a different day. The algorithms were evaluated on larger galleries and progress from the previous FERET evaluation in 1994 was measured. This evaluation consisted of a single test that measured identification performance from a gallery of 817 individuals.

The third FERET test was administered in September 1996 and March 1997. The design of this evaluation was more complex than the first two evaluations, and allowed for more detailed performance characterization of face recognition systems. One of the main goals of FERET96 was to measure the improvements of performance of face recognition algorithms on different probe and gallery image sets since the last two FERET evaluations. FERET96 tried to measure the effect of pose variation and digital modification of the probe images on performance.

1.1.2 Facial Recognition Vendor Test

From 2000, the FERET program also began to evaluate commercial face recognition systems. The Facial Recognition Vendor Test 2000 (FRVT 2000) [24] was conducted to assess the technical capabilities of commercial face recognition systems. FRVT 2000 conducted two kinds of tests:

1. Recognition Performance Test - technology evaluation.

2. Product Usability Test - limited scenario evaluation.

The recognition performance test was conducted to evaluate the technical performance of the face recognition systems. The tests were conducted on the standardized FERET
database. The change in performance due to the following factors was quantified using CMS and ROC curves:

2. Distance: Estimate effect if position subject at varying distance from a fixed camera.
3. Expression: Evaluate performance when comparing images of the same person with different facial expression.
4. Illumination: Analyze effect of changes of subject illumination.
5. Media: Estimate effect of comparing images stored on different media.
6. Pose: Evaluate performance as viewpoint from which facial images are taken changes.
7. Image resolution: Evaluate performance as image resolution is varied.
8. Temporal: Analyze effect of time delay between first and subsequent capture of facial images.

The Product Usability test was a limited scenario evaluation. Its scenario for the test was access control to live subjects. This test was carried out though some of the participant face recognition systems were not designed for access control applications. During the product usability test some of the parameters that were varied are the start distance, behavior mode and backlighting. The test subjects performed each test in the cooperative, repeatable and indifferent behavior modes.

The next face recognition vendor test was conducted in 2002 (FRVT 2002) [23]. FRVT 2002 consisted of two subtests namely

1. High Computational Intensity (HCInt) test.
2. Medium Computational Intensity (MCInt) test.

The HCInt test was designed to evaluate the performance of the face recognition systems on a very large database. The MCInt test was designed to evaluate the capability of the face
recognition system to perform the face recognition task with several different formats of imagery (still and video) under varying conditions like indoor lighting and outdoor lighting.

Performance of the face recognition systems was also measured on watch lists. Watch list is a list of the subjects that the face recognition system is on the look out for. When a probe is presented to the face recognition system then the face recognition system checks to see if it is present on the watch list. The performance of this application is characterized by the watch list ROC that plots to trade-off between detection and identification rate and false alarm rate.

One of the new features of FRVT 2002 was that it computed the variance in performance for multiple probe and gallery combinations. Error ellipses were used in the ROC curves to estimate the range of performance. The error ellipses are measured using disjoint galleries and probe sets. This avoids the issues with resampling techniques. Error ellipses are a measure of variance and unlike confidence intervals they do not provide error bounds on the ROC. Some of the covariates that were explored for the first time by FRVT 2002 are sex and age of an individual.

1.2 Evaluating Statistical Significance of the Results

The FERET protocol was a major step in the standardizing the evaluation of face recognition algorithms. However the FERET protocol stopped short of addressing the critical question of statistical variability [26]. Sometimes not all the measured differences between the performance of algorithms are statistically significant. Some of the differences also occur by chance. So it was essential to develop sound techniques to study the statistical significance of the measured differences. The FERET protocol did not establish a common means of testing when the difference between two curves is significant. So methods need to be developed to address this issue.

Micheals et al [28] try to derive the mean and standard deviation estimates for recognition rates at different ranks. By using stratified sampling and a statistical technique called Balanced Repeated Resampling (BRR) they generate standard error bars for CMS curves. They used their technique to compare the difference between PCA algorithm and
two algorithms from the Visionics FaceIt SDK on the FERET data set. One of the simplest technique for evaluating statistical significance is the McNemar’s test [25]. If two trained algorithms are tested with the same gallery and probe images then this test would be very appropriate to compare the statistical significance of the results. Beveridge et al also permuted the probe and the gallery images to verify the statistical significance of the results and thus generated CMS curves with error bars indicating confidence intervals [25].

Jonathon Phillips et al [15] explored the issue of whether human subjects and face recognition algorithms both find the same faces identical. The results were represented using biplots. Leigh et al [10] used Phi-PIT transformations to represent the similarity matrixes using box plots and analyze the results using ANOVA. The box plots help in characterizing the distribution of the data set. Rukhin et al [31] used partial rank correlations to correlate the performance of face recognition algorithms.

1.2.1 Significance of the Affine Approximating Algorithm

The affine approximation algorithm is essentially a tool for carrying out the technology evaluation of face recognition algorithms at a deep level. The PCA face recognition algorithm has frequently been used as a baseline algorithm to evaluate the performance of other face recognition algorithms [16, 21]. So far the comparison with the PCA algorithm has only been made in the identification and the verification scenarios using final performance scores and statistical analysis. The affine approximation algorithm takes a step further by not treating the face recognition algorithm as a black box. It tries to generate an affine transform that attempts to transform the eigen space of the PCA algorithm so that it matches the results of the face recognition algorithm as closely as possible. The affine transform will be an identity matrix if the input face recognition is the PCA algorithm. The closer the affine transform is to an identity transform the closer the input face recognition algorithm is to the PCA algorithm. Subsequent sections deal with the error measures and techniques used to compare the similarity matrix obtained by the face recognition algorithm and the affine approximating algorithm.
CHAPTER 2
AFFINE APPROXIMATION ALGORITHM

Figure 2.1. Problem Definition. Find matrix $A$ such that the Euclidean distances between transformed images i.e. $(A) \text{ (Input Images)}$ are equal to the given distances.

This chapter (Figure 2.1) outlines the algorithm that is used to approximate the face recognition algorithm. Every face recognition algorithm can output a similarity matrix between the images in the gallery and the probe set. The affine approximation algorithm uses this similarity matrix to produce an optimal affine transformation. The affine transformation matrix is a closed form solution. This affine transformation can translate, rotate, stretch and shear the eigen space of the PCA algorithm such that when the images can be embedded in this space the Euclidean distances between the images matches the similarity scores produced by the face recognition algorithm. If the input face recognition algorithm is the PCA algorithm then the affine transformation will be an identity matrix. The closer the affine approximation algorithm is to an identity matrix the closer we can say that the behavior of an algorithm is to the PCA algorithm. In this manner any input face recognition algorithm can be compared with the baseline PCA algorithm.
The following notation will be used to describe the linear transform strategy.

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 $d_{ij}^A$ 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.

\[ \vec{y}_i = A\vec{x}_i \] (2.1)

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

\[ d^E(\vec{y}_i, \vec{y}_j) = \sum_{k=1}^{M} (\vec{y}_i(k) - \vec{y}_j(k))^2 = (\vec{y}_i - \vec{y}_j)^T(\vec{y}_i - \vec{y}_j) \] (2.2)

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

\[ d^E(\vec{y}_i, \vec{y}_j) = d_{ij}^A \] (2.3)

\[ d^E(\vec{y}_i, \vec{y}_j) = (\vec{y}_i - \vec{y}_j)^T(\vec{y}_i - \vec{y}_j) \\
= (A\vec{x}_i - A\vec{x}_j)^T(A\vec{x}_i - A\vec{x}_j) \\
= (A(\vec{x}_i - \vec{x}_j))^T(A(\vec{x}_i - \vec{x}_j)) \\
= (\vec{x}_i - \vec{x}_j)^T(A^T A)(\vec{x}_i - \vec{x}_j) \] (2.4)

(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 $A^T A = AA^T = I$, 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

$$
\begin{align*}
    d^E(\vec{y}_i, \vec{y}_j) &= \vec{e}_{ij}^T B \vec{e}_{ij} \\
    &= \sum_{k=1}^{N^2} \sum_{l=1}^{N^2} B(k, l) \vec{e}_{ij}(k) \vec{e}_{ij}(l)
\end{align*}
$$

(2.5)

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 $N^2(N^2+1)/2$ sized column vector by scanning the lower triangular entries (including the diagonal) of $B$. Thus,

$$
\vec{b}(\frac{k(k+1)}{2} + l) = B(k, l), \text{ for } l \leq k, k = 1, \cdots, N^2
$$

(2.6)

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

$$
\vec{\psi}_{ij}(\frac{k(k+1)}{2} + l) = \begin{cases} 
    \vec{e}_{ij}(k)^2 & \text{for } k = l \\
    2\vec{e}_{ij}(k)\vec{e}_{ij}(l) & \text{for } l \leq k
\end{cases}
$$

(2.7)

Using the above equation

$$
\begin{align*}
    d^E(\vec{y}_i, \vec{y}_j) &= \vec{\psi}_{ij}^T \vec{b} \\
    \vec{b} &\text{ should be determined such that}
\end{align*}
$$

(2.8)

(2.9)

for every pair of images. These $\frac{K(K-1)}{2}$ equations can be compactly expressed in matrix notion as follows

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

(2.10)
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 $\vec{\psi}_{ij}$. 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 $B = A^T A$. 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. 2.5 is 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

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

(2.11)

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.

To summarize, the steps are

1. Form the $\frac{K(K-1)}{2} \times \frac{N^2(N^2+1)}{2}$ sized matrix $\psi$ from the input images as described above. The constraint is that $K \geq N^2 + 1$.

2. Form the column vector $\vec{d}^A$ from the given distances.

3. Find $B$ by solving the linear equation $\psi^T \vec{b} = d^A$, where $\vec{b}$ is related to $B$ as described above.
Figure 2.2. Flow Chart of the Affine Approximation Algorithm.

Find $\psi$ such that

$$
\tilde{e}_{ij}(\frac{k(k+1)}{2} + l) = \begin{cases} 
\tilde{e}_{ij}(k)^2 & \text{for } k = l \\
2\tilde{e}_{ij}(k)\tilde{e}_{ij}(l) & \text{for } l \leq k
\end{cases}
$$

Form column vector of the given distances $d_A$

Solve

$$
\psi^T \tilde{e} = \tilde{d}
$$

Find $B$ such that

$$
\tilde{B}(\frac{k(k+1)}{2} + l) = B(k, l), \text{ for } l \leq k, k = 1, \ldots, N^2
$$

Find eigen vectors $(u_l)$ & eigen values $(\lambda_l) B$

Form $A$ such that the $i$-th row of $A$ is $\sqrt{\lambda_i} u_i^T$. 
4. Find the eigenvectors ($\vec{u}_i$) and eigenvalues ($\lambda_i$) of $B$

5. Form $A$ such that the $i$-th row of $A$ is $\sqrt{\lambda_i} \vec{u}_iT$.

One of the concerns with this approach is the requirement that the number of images available ($K$) be greater than size of the image ($N^2$), the dimension of the pixel-based space of the raw images. This requirement can be relaxed if we first perform a principal component analysis (PCA), which preserves Euclidian distances, on the original pixel-based image space, to arrive at a smaller subspace, say, of size $P$ dimensions with $P << N$. In that case the requirement would be that the number of images available be larger than this number of dimension.

So the overall steps of the algorithm is as illustrated in Figure 2.2. The estimated matrixes are $T_{PCA}$ that captures the rigid transform and $A$ captures the non-rigid part. The overall affine transform is $AT_{PCA}$. If $A$ is the identity matrix then we get the plain PCA algorithm. The affine transform matrix shears and stretches the eigen space of the algorithm so that it can embed the images such that the Euclidean distance between them equals the original similarity scores.
CHAPTER 3
TEST ALGORITHMS

Three standard algorithms were chosen to test the performance of the affine approximation strategy. The algorithms chosen were:

1. Principal Component Analysis (PCA) algorithm [5].

2. Principal Component Analysis (PCA) coupled with the Linear Discriminant Analysis (LDA) algorithm [1].

3. Bayesian intrapersonal and extrapersonal classifier [3].

An implementation of the algorithms was available from [29]. A brief description of each of these algorithms is as follows:

3.1 Principal Component Analysis (PCA) Algorithm

Kirby et al developed a technique to represent faces by using the Karhunen-Loève projection [4]. This technique is known as Principal Component Analysis (PCA). Essentially PCA is a statistical dimensionality reduction method, which produces the optimal linear least squared decomposition of a training set. This technique was extended by Turk et al for the task of face recognition [5].

In the PCA algorithm, first each training image is unrolled into a vector of n pixel values. Then the mean image of the training set is calculated and it is subtracted from each training image. So the resulting training images are "mean centered". All the mean centered images are placed as a column of a matrix, say $M$. The covariance matrix

$$\Omega = MM^T$$  \hspace{1cm} (3.1)
The covariance matrix characterizes the distribution of the images in $\mathbb{R}^n$. The eigen vectors of the covariance matrix $\Omega$ form the subspace in which the gallery and the probe images will be embedded for comparison. The eigenvectors are normalized so that they become orthonormal. The eigenvalues of the eigen vectors are inversely proportional to the amount of variance each eigen vector represents. The eigen vector with the highest eigenvalue corresponds to the direction of the maximum variance and so on. These eigen vectors are the Principal Components of the training images.

During the testing phase, the gallery and the probe images are mean centered and projected onto the eigen space. The distance between the gallery and the probe images in the eigen space represents their similarity score. Difference distance measures can be used to compute the distances between the gallery and the probe images. Two distances measures were choosen to test the affine approximation algorithm. That are the Euclidean distance measure and the cosine distance measure. As anticipated the affine transform matrix is an identity matrix when the distances between the gallery and the probe images are computed using the Euclidean distance measure.

3.2 Linear Discriminant Analysis (LDA) Algorithm

The Linear Discriminant Analysis algorithm was developed by Zhao and Chellapa [1]. The LDA algorithm is based on Fischer’s Linear Discriminants. Essentially LDA tries to produce an optimal linear discriminant function that emphasizes the differences between classes and minimizes the differences within classes. In face recognition each class refers to the set of images of each subject. So while training the LDA algorithm more than one sample of each subject is required. Initially PCA is performed to reduce the dimensionality of the feature vectors. Then the LDA is performed so that the class distinguishing features are preserved. The PCA and LDA basis vectors are multiplied to produce the LDA transformation matrix. The gallery and the probe images are projected onto the LDA space and are compared as before.
3.3 Bayesian Intrapersonal and Extrapersonal Classifier

The Bayesian intrapersonal and extrapersonal classifier was developed by Moghaddam and Pentland [3]. This algorithm uses a probabilistic measure of similarity for comparison. There are two variants of the algorithms

1. Maximum Likelihood (ML) Classifier.


This algorithm tries to model two mutually exclusive classes of differences: intra-personal and the extra-personal differences. The intra personal differences are variations in the appearance of the same individual due to different expressions, lighting changes etc. The extra personal differences are the variations in appearances between different individuals. The MAP classifier uses the intra personal and extra personal classifier whereas the ML classifier uses only the intra personal differences. Results show that the ML classifier performs just as well as the MAP classifier [3]. Also the ML classifier is computationally less expensive. In this study the ML classifier was used for comparing with the affine approximation algorithm.
CHAPTER 4
EXPERIMENTAL SETUP

This chapter describes the data set and the training scenarios that were used to compare the face recognition algorithm with the affine approximation algorithm.

4.1 Data Description

Three disjoint images sets $C_1$, $C_2$ and $C_3$ were selected from the FERET database [16, 18]. All the three sets, $C_1$, $C_2$ and $C_3$ consisted of images of the type fa (regular facial expression) and fb (alternate facial expression of the subject taken with the same lighting conditions). Sets $C_1$ and $C_2$ consisted of 100 images of 25 subjects (i.e) 4 images per subject. Set $C_3$ consisted of 600 images of 300 subjects (i.e) 2 images per subject. Sets $C_1$ and $C_2$ were used to train the face recognition algorithms and the affine approximation algorithm while set $C_3$ was used as a validation set to compare the similarity matrix generated by the face recognition algorithm and the affine approximation algorithm. The validation set $C_3$ consisted of images of type fa and fb. The corresponding fb image for each fa image was taken on the same day. All the images were preprocessed using the normalization code developed at NIST. The images were spatially normalized such that the eyes were always placed at fixed points in the imagery based upon a ground truth file of eye coordinates provided with the FERET data. The images were cropped to a standard size of 150 by 130 pixels. Pixels not lying within an oval shaped face region were masked out. Then the pixel values were histogram equalized and then shifted and scaled such that the mean value of all the pixels in the face region is zero and the standard deviation is one.

4.2 Training Scenarios

Two training scenarios for the algorithms were considered:
Figure 4.1. Training Scenario 1: Block Diagram. Face recognition algorithm and affine approximation algorithm trained on the same image set.

Figure 4.2. Training Scenario 2: Block Diagram. Face recognition algorithm and affine approximation algorithm trained on different image sets.
1. Both the face recognition algorithm and the affine approximation algorithm were trained using the same set (either $C_1$ or $C_2$).

2. The face recognition algorithm was trained on set $C_1$ and the affine approximation algorithm was trained on set $C_2$.

The Linear Discriminant Analysis classifier algorithm and the Bayesian interpersonal and intrapersonal classifier algorithm both first perform a PCA to reduce the dimensionality of the features. In the first training scenario the dimensionality of the PCA step in the affine approximation algorithm is maintained constantly at 60 dimensions. The results from the first training scenario significantly improved when the dimensionality of the affine approximation was gradually increased to match the dimensionality of PCA step in the face recognition algorithms. The best results were obtained when the dimensionality of the affine approximation was made equal to the dimensionality of the PCA space. The results did not show significant difference when the dimensionality of the affine approximation was made greater than the dimensionality of the PCA space. So in the second training scenario the dimensionality of the PCA step was always made equal to the dimensionality of the affine approximation algorithm. The second scenario is a more stringent way of comparing a face recognition algorithm with the PCA algorithm. The second scenario seeks to compare how close an already trained face recognition algorithm is with the PCA algorithm. In the first scenario both the face recognition algorithm and the affine approximation algorithm are trained on the same image set. In the second training scenario the face recognition algorithm and the affine approximation algorithm are trained on different image sets.
CHAPTER 5
ANALYSIS OF RESULTS

This chapter looks into some of the techniques that have been used to compare the similarity matrix generated by the face recognition algorithm and the affine approximation algorithm and then presents the results. Both the face recognition algorithm and the affine approximation algorithm output a similarity matrix. The similarity matrix indicates the similarity score between each pair of image in the test set $C_3$.

The raw similarity scores produced by the face recognition algorithm and the affine approximation algorithm are compared directly by visualizing the distribution of the values in the similarity matrixes using images, by studying the histogram of the matrix obtained by the difference between the similarity matrix generated by the affine approximation algorithm and the face recognition algorithm and by using standard gross error measures like average RMS error and Stress-1.

Alternatively the performance of the face recognition algorithm and the affine approximation algorithm in the identification and the verification scenarios are also used as an index to judge the performance of the two algorithms. When the input face recognition algorithm is the PCA face recognition algorithm then the affine transformation matrix will be an identity matrix. So the closer the affine approximation matrix is to an identity matrix, the closer the face recognition algorithm is to the PCA algorithm. So techniques are also used to visualize the affine transformation matrix. The subsequent sections will elaborate more on these techniques.

5.1 Visualization of Distance Matrixes

In Figures 5.1 to 5.4 the distance matrixes generated by the affine approximation algorithm and the face recognition algorithm are directly compared. The individual distance
matrixes along with the error matrixes are visualized as images with red denoting larger values than blue mapped according to the legend bar alongside each image. The histogram of the difference matrixes clearly shows that the distance matrix produced by the affine approximation algorithm closely matches the distance matrix produced by the face recognition algorithm. This is more pronounced in the first training scenario then in the second training scenario. This may be probably because of the difference in the training sets.

In training scenario 1, the dimensionality of the PCA step in all the three algorithms is made 60. In the PCA face recognition algorithm using the Euclidean distance measure, when the dimensionality of the affine space becomes equal to the dimensionality of the PCA space (ie 60 dimensions) the similarity matrix produced by the affine approximation algorithm becomes identical to the similarity matrix produced by the PCA algorithm (see Figure 5.1). In the PCA face recognition algorithm using the cosine distance measure, the histogram shows that there is a significant difference between the distance matrix generated by the affine approximation algorithm and the PCA algorithm (see Figure 5.2). In the case of the LDA algorithm the best results are obtained when the dimensionality of the affine space becomes equal to the dimensionality of the LDA space (ie 24 dimensions). The Bayesian algorithm does not use an explicit multidimensional space to classify the images. But the results become better as the dimensionality of the affine space is gradually increased to match the dimensionality of the PCA space. Significant differences in the results are observed between the training scenario 1 and training scenario 2 for the PCA and the LDA face recognition algorithms. But for the Bayesian face recognition algorithm changes in the training set does not seem to make much difference (see Figure 5.4).

5.2 Gross Error Measures

The normalized average RMS error and Stress-1 [32], which is a standard error measure used in MDS, were used to quantify the difference between the similarity matrix generated by the face recognition algorithm and the distance matrix generated by the affine
Figure 5.1. Visualization of Similarity Matrix for the PCA Algorithm Using the Euclidean Distance Measure. Values are in generic units. (a) Similarity matrix produced by PCA algorithm (b) Similarity matrix produced by Affine approximation algorithm (c) Difference between the two similarity matrixes (d) Histogram of the difference matrix.
Figure 5.2. Visualization of Similarity Matrix for PCA Algorithm Using the Cosine Distance Measure. Values are in generic units. (a) Similarity matrix produced by PCA algorithm (b) Similarity matrix produced by Affine approximation algorithm (c) Difference between the two similarity matrixes (d) Histogram of the difference matrix.
Figure 5.3. Visualization of Similarity Matrix for the LDA Algorithm. Values are in generic units. (a) Similarity matrix produced by the LDA algorithm (b) Similarity matrix produced by Affine approximation algorithm (c) Difference between the two similarity matrixes (d) Histogram of the difference matrix.
approximation algorithm. The normalized average RMS error is defined as follows:

\[
\frac{1}{\max(D)} \sqrt{\frac{\sum (D_{ij} - D'_{ij})^2}{N^2}}
\]  

(5.1)

where \(D\) is the distance matrix obtained using the face recognition algorithm, \(D'\) is the distance matrix obtained using the affine approximation algorithm and \(\max(D)\) is the maximum similarity score of the distance matrix \(D\).

Stress-1 is defined as follows:

\[
\sigma = \sqrt{\frac{\sum (D_{ij} - D'_{ij})^2}{\sum D_{ij}^2}}
\]  

(5.2)

Figure 5.5 shows the plot of the average RMS error and Figure 5.6 shows the stress plot for training scenario 1. From Figure 5.5 and Figure 5.6 it can be observed that once the dimensionality of the affine approximation algorithm becomes equal to the dimensionality of the face space used by the PCA (Euclidean distance measure) and the LDA algorithms the error values become constant. No such phenomena is observed in the Bayesian intrapersonal and extrapersonal classifier algorithm. The error values for the PCA face recognition algorithm using the cosine distance measure are the highest and they keep increasing as the dimensionality of the affine space increased. This is because the image of the distance matrix generated by the PCA algorithm using the cosine distance measure shows that the algorithm tends to cluster the images into groups whereas the affine approximation algorithm tends to distribute the images more evenly in the affine space (see figure 6.2). But the CMS and the ROC curves show that the performance of the affine approximation algorithm is better than the PCA face recognition algorithm using the cosine distance measure and this performance improves as the dimensionality of the affine space is increased. Figure 5.7 and Figure 5.8 show the Average RMS error values and Stress 1 values for training scenario 2. In this case, the dimensionality of the affine space is made equal to the dimensionality of the PCA space in each of the face recognition algorithms.
Figure 5.4. Visualization of Similarity Matrix for the Bayesian Algorithm. Values are in generic units. (a) Similarity matrix produced by the Bayesian algorithm (b) Similarity matrix produced by Affine approximation algorithm (c) Difference between the two similarity matrixes (d) Histogram of the difference matrix. The difference between training scenario 1 and training scenario 2 does not seem to be significant.
Figure 5.5. Training Scenario 1: Plot of RMS Error. Dimensionality of the PCA step in all the three algorithms was set to 60 dimensions.

Figure 5.6. Training Scenario 1: Stress Plot. Dimensionality of the PCA step in all the three algorithms was set to 60 dimensions.
Figure 5.7. Training Scenario 2: Plot of RMS Error. Dimensionality of the PCA step in all the three algorithms was made equal to the dimensionality of the affine space.

Figure 5.8. Training Scenario 2: Stress Plot. Dimensionality of the PCA step in all the three algorithms was made equal to the dimensionality of the affine space.
5.3 Analysis of Affine Approximation Algorithm

The affine approximation algorithm uses the similarity matrix generated by a face recognition algorithm to come up with an optimal affine transformation. The affine transformation will try to stretch and shear the eigen space of the PCA algorithm so that the images can be embedded in the space such that the Euclidean distance between the embedded images matches the similarity scores generated by the face recognition algorithm. If the input face recognition algorithm is the PCA algorithm then the affine transformation we would expect the affine transformation to be an identity matrix. This is seen in Figure 5.9.

It is worthwhile to analyze the affine transformation matrix. The affine transformation matrix has been visualized as an image with the rows and columns permuted so that the diagonal elements contain the most dominant element. Taking a clue from discrete dynamical systems, the eigen values of A were computed. If the eigen value $\lambda_i < 1$ then it means that the affine space is getting compressed along the $i^{th}$ dimension. If $\lambda_i > 1$ then it means that the affine space is getting stretched along the $i^{th}$ dimension. So the overall distribution of the eigen values would give an idea as to what the affine transformation is doing to the eigen space, whether the eigen space is being stretched, sheared or compressed. Figures 5.9 to 5.12 show the the affine transform for all the four algorithms for the training scenario 1 and figures 5.13 to 5.16 show the affine transform for the training scenario 2.
Figure 5.9. Training Scenario 1: Visualization of Affine Transformation Matrix for the PCA Algorithm Using the Euclidean Distance Measure. Values are in generic units. (a) Affine transformation matrix with the rows and columns rearranged so that the diagonal elements contain the largest value. In this case the affine transformation matrix is an identity matrix. (b) The plot of the eigen values of the affine transformation matrix shows that the eigen space has not been stretched or sheared.

Figure 5.10. Training Scenario 1: Visualization of Affine Transformation Matrix for the PCA Algorithm Using the Cosine Distance Measure. Values are in generic units. (a) Affine transformation matrix with the rows and columns rearranged so that the diagonal elements contain the largest value. (b) The plot of the eigen values of the affine transformation matrix shows that the eigen space has been compressed by the affine transform.
Figure 5.11. Training Scenario 1: Visualization of Affine Transformation Matrix for the LDA Algorithm. Values are in generic units. (a) Affine transformation matrix with the rows and columns rearranged so that the diagonal elements contain the largest value. (b) The plot of the eigen values of the affine transformation matrix shows that the eigen space has been compressed by the affine transform.

Figure 5.12. Training Scenario 1: Visualization of Affine Transformation Matrix for the Bayesian Algorithm. Values are in generic units. (a) Affine transformation matrix with the rows and columns rearranged so that the diagonal elements contain the largest value. (b) The plot of the eigen values of the affine transformation matrix shows that the eigen space has been compressed by the affine transform.
Figure 5.13. Training Scenario 2: Visualization of Affine Transformation Matrix for the PCA algorithm Using the Euclidean Distance Measure. Values are in generic units. (a) Affine transformation matrix with the rows and columns rearranged so that the diagonal elements contain the largest value. (b) The plot of the eigen values of the affine transformation matrix shows that the eigen space has been compressed by the affine transform.

Figure 5.14. Training Scenario 2: Visualization of Affine Transformation Matrix for the PCA Algorithm Using the Cosine Distance Measure. Values are in generic units. (a) Affine transformation matrix with the rows and columns rearranged so that the diagonal elements contain the largest value. (b) The plot of the eigen values of the affine transformation matrix shows that the eigen space has been compressed by the affine transform.
Figure 5.15. Training Scenario 2: Visualization of Affine Transformation Matrix for the LDA Algorithm. Values are in generic units. (a) Affine transformation matrix with the rows and columns rearranged so that the diagonal elements contain the largest value. (b) The plot of the eigen values of the affine transformation matrix shows that the eigen space has been compressed by the affine space.

Figure 5.16. Training Scenario 2: Visualization of Affine Transformation Matrix for the Bayesian Algorithm. Values are in generic units. (a) Affine transformation matrix with the rows and columns rearranged so that the diagonal elements contain the largest value. (b) The plot of the eigen values of the affine transformation matrix shows that the eigen space has been compressed by the affine space.
5.4 Eigen Values of B Matrix

The affine approximation matrix A is obtained by the eigen vector decomposition of the $B$ matrix. So it is worthwhile to study how the eigen values of the $B$ matrix decrease as the dimensionality of the affine space is increased. Figures 5.17 to 5.24 show the plot of the eigen values of $B$ matrix. In training scenario-1 the eigen values of $B$ matrix are nearly unity until the dimensionality is less than 60 and then after crossing 60 the eigen values become zero. In the LDA algorithm the eigen values become zero once the dimensionality of the affine space crosses the dimensionality of the LDA space ie 24 dimensions. For the other algorithms the eigen values of $B$ matrix become nearly zero as the dimensionality is increased further and further. Thus the eigen values of $B$ can be used for determining the optimal dimensionality of the affine space.

![Eigen Value Index of Eigen Vector](image)

Figure 5.17. Training Scenario 1: Eigen Values for the PCA Algorithm Using the Euclidean Distance Measure. The eigen values are nearly unity until the dimensionality is less than 60 and after that they become zero.
Figure 5.18. Training Scenario 1: Eigen Values for the PCA Algorithm Using the Cosine Distance Measure.
Figure 5.19. Training Scenario 1: Eigen Values for the LDA Algorithm. The eigen values become zero when the dimensionality crosses the dimensionality of the LDA space.

Figure 5.20. Training Scenario 1: Eigen Values for the Bayesian Algorithm.
Figure 5.21. Training Scenario 2: Eigen Values for the PCA Algorithm Using the Euclidean Distance Measure.
Figure 5.22. Training Scenario 2: Eigen Values for the PCA Algorithm Using Cosine Distance Measure.

Figure 5.23. Training Scenario 2: Eigen Values for the LDA Algorithm.
Figure 5.24. Training Scenario 2: Eigen Values for the Bayesian Algorithm.
5.5 Performance of Identification and Verification Scenarios

The previous subsection compares the raw similarity scores. However, it is necessary to look into the performance of any face recognition algorithm in the identification and verification scenario. So the distance matrixes from the affine approximation algorithm and face recognition algorithm were tested in the identification and verification scenarios also.

The validation set $C_3$ was partitioned into two disjoint subsets to form the gallery and probe sets. The gallery set consisted of all the images of type fa(normal expression) in the validation set $C_3$ and the probe set consisted of all the images of type fb(different expression, same lighting and same day) in the validation set. The performance of the affine approximation algorithm and the face recognition algorithms for the identification and verification scenarios was characterized by using standard Cumulative Match Score (CMS) curve and Receiver Operator Characteristics (ROC) curve.

Figures 5.25, 5.27, 5.29 and 5.31 show the CMS curves for the three algorithms for training scenario 1 while figures 5.26, 5.28, 5.30 and 5.32 shows the ROC curves. From the CMS and ROC curves it is evident that as the dimensionality of the affine approximation algorithm is increased the closeness between the CMS curve of the affine approximation algorithm and the face recognition algorithm also increases. The ROC curves also show that the affine approximation algorithm approximates the face recognition algorithms very well. Figures 5.33, 5.35, 5.37 and 5.39 show the CMS curves for training scenario 2. The CMS curves for the second training scenario are not as good as the CMS curves for the first training scenario because of the effect of training the algorithms on different sets. Figures 5.34, 5.36, 5.38 and 5.40 show the ROC curves for the second training scenario. The ROC curves between the affine approximation algorithm and the face recognition algorithms for the second training scenarios are very close.

5.6 McNemar’s Test

Beveridge et al [25] demonstrated the appropriateness of using the McNemar’s test for comparing two identification rates. McNemar’s test is very suitable for paired data. If the
Figure 5.25. Training Scenario 1: CMC Curve for PCA Algorithm Using the Euclidean Distance Measure.

Figure 5.26. Training Scenario 1: ROC Curve for PCA Algorithm Using the Euclidean Distance Measure.
Figure 5.27. Training Scenario 1: CMC Curve for PCA Algorithm Using Cosine Distance Measure.

Figure 5.28. Training Scenario 1: ROC Curve for PCA Algorithm Using Cosine Distance Measure.
Figure 5.29. Training Scenario 1: CMC Curve for LDA Algorithm.

Figure 5.30. Training Scenario 1: ROC Curve for LDA Algorithm.
Figure 5.31. Training Scenario 1: CMC Curve for Bayesian Algorithm.

Figure 5.32. Training Scenario 1: ROC curve for Bayesian Algorithm.
Figure 5.33. Training Scenario 2: CMC Curve for PCA Algorithm Using the Euclidean Distance Measure.

Figure 5.34. Training Scenario 2: ROC Curve for PCA Algorithm Using the Euclidean Distance Measure.
Figure 5.35. Training Scenario 2: CMC Curve for PCA Algorithm Using Cosine Distance Measure.

Figure 5.36. Training Scenario 2: ROC Curve for PCA algorithm Using Cosine Distance Measure.
Figure 5.37. Training Scenario 2: CMC Curve for LDA Algorithm.

Figure 5.38. Training Scenario 2: ROC Curve for LDA Algorithm.
Figure 5.39. Training Scenario 2: CMC Curve for Bayesian Algorithm.

Figure 5.40. Training Scenario 2: ROC Curve for Bayesian Algorithm.
gallery and the probe set used for testing two algorithms is identical, then McNemar’s test can be applied to study if the differences in performance between the algorithms are statistically significant. In such a situation there maybe four outcomes for each comparision:

1. Both the algorithms succeed - SS.
2. Both the algorithms fail - FF.
3. Algorithm A succeeds but algorithm B fails - SF.
4. Algorithm A fails but algorithm B succeeds - FS.

McNemar’s test discards the first two outcomes, SS and FF. The null hypothesis $H_0$, is that the probability of observing SF is equal to the probability of observing FS. The alternative hypothesis is that the probability of observing SF is not equal to the probability of observing FS. Let $n_{SF}$ denote the number of times SF is observed and $n_{FS}$ denote the number of times FS.

$$P[n_{SF}] = P[n_{FS}] = \sum_{i=0}^{n_{FS}} \frac{n!}{i!(n-i)!}0.5^n$$

(5.3)

where $n = n_{SF} + n_{FS}$. This probability is the p value for accepting the null hypothesis, $H_0$.

Table 5.1 shows the results of McNemar’s test for the first training scenario and tables 5.2 and 5.3 shows the results for the second training scenario. In the first training scenario, when the dimensionality of the affine approximation algorithm is lesser than the dimensionality of the eigen space or the LDA space in the then the null hypothesis $H_0$ becomes true only at higher ranks. Such behavior can also be observed in the case of the Bayesian algorithm even though it does not use a multidimensional space to classify to probe and gallery images. Once the dimensionality of the affine approximation algorithm and the eigen space or the LDA space become equal than $H_0$ becomes true at rank 1. However in the second training scenario the differences between $P_{SF}$ and $P_{FS}$ for all the algorithms except the PCA face recognition algorithm using the cosine distance measure become significant only at higher ranks. This is probably because the algorithms are trained on two different sets.
Table 5.1. McNemar’s Test for Training Scenario 1 at Rank 1. SF denote the number of times the face recognition algorithm succeeds in recognizing the image but the affine approximation algorithm fails. FS denote the number of times the affine approximation algorithm succeeds in recognizing the image but the face recognition algorithm fails. RRF denotes the recognition rate for the face recognition algorithm and RRA denotes the recognition rate for the affine approximation algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SF</th>
<th>FS</th>
<th>p value</th>
<th>RRF</th>
<th>RRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA (Euclidean)</td>
<td>0</td>
<td>0</td>
<td>not applicable</td>
<td>72%</td>
<td>72%</td>
</tr>
<tr>
<td>PCA (Cosine)</td>
<td>12</td>
<td>12</td>
<td>0.8383</td>
<td>69.67%</td>
<td>69.67%</td>
</tr>
<tr>
<td>LDA</td>
<td>7</td>
<td>1</td>
<td>0.0771</td>
<td>79.33%</td>
<td>77.33%</td>
</tr>
<tr>
<td>Bayesian</td>
<td>14</td>
<td>8</td>
<td>0.2864</td>
<td>83%</td>
<td>81%</td>
</tr>
</tbody>
</table>

Table 5.2. McNemar’s Test for Training Scenario 2 at Rank 1. SF denote the number of times the face recognition algorithm succeeds in recognizing the image but the affine approximation algorithm fails. FS denote the number of times the affine approximation algorithm succeeds in recognizing the image but the face recognition algorithm fails. RRF denotes the recognition rate for the face recognition algorithm and RRA denotes the recognition rate for the affine approximation algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SF</th>
<th>FS</th>
<th>p value</th>
<th>RRF</th>
<th>RRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA (Euclidean)</td>
<td>25</td>
<td>1</td>
<td>0.0001</td>
<td>72.67%</td>
<td>64.67%</td>
</tr>
<tr>
<td>PCA (Cosine)</td>
<td>13</td>
<td>12</td>
<td>1.0</td>
<td>71.67%</td>
<td>71.33%</td>
</tr>
<tr>
<td>LDA</td>
<td>28</td>
<td>11</td>
<td>0.0104</td>
<td>76.67%</td>
<td>71%</td>
</tr>
<tr>
<td>Bayesian</td>
<td>21</td>
<td>5</td>
<td>0.0033</td>
<td>83%</td>
<td>77.67%</td>
</tr>
</tbody>
</table>

Table 5.3. McNemar’s Test for Training Scenario 2. The rank column denotes the rank from which the difference becomes insignificant. RRF denotes the recognition rate for the face recognition algorithm and RRA denotes the recognition rate for the affine approximation algorithm at that rank.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rank</th>
<th>SF</th>
<th>FS</th>
<th>p value</th>
<th>RRF</th>
<th>RRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA (Euclidean)</td>
<td>7</td>
<td>12</td>
<td>5</td>
<td>0.1456</td>
<td>83.67%</td>
<td>81.3%</td>
</tr>
<tr>
<td>PCA (Cosine)</td>
<td>1</td>
<td>13</td>
<td>12</td>
<td>1.0</td>
<td>83.67%</td>
<td>81.3%</td>
</tr>
<tr>
<td>LDA</td>
<td>4</td>
<td>12</td>
<td>2</td>
<td>0.0704</td>
<td>96.3%</td>
<td>85%</td>
</tr>
<tr>
<td>Bayesian</td>
<td>11</td>
<td>7</td>
<td>2</td>
<td>0.1824</td>
<td>96.3%</td>
<td>94.67%</td>
</tr>
</tbody>
</table>
5.7 Difference Caused by Training Scenarios

All the results show that the performance in the first training scenario is always better than the second training scenario. In the first training scenario both the algorithms are trained with the same data set and in the second training scenario they are trained with different data sets. Figure 5.41 and Figure 5.42 show the top eigen faces of training set $C_1$ and training set $C_2$ respectively. The top eigen faces for both the sets are very different. Probably this difference can be reduced by using a larger training set. But we were unable to explore this since the computational cost and time becomes prohibitively expensive as the training set size increases.
CHAPTER 6
CONCLUSION

We have presented a very novel method of evaluating face recognition algorithms at a much deeper level. The affine approximation algorithm uses the similarity of any input algorithm to generate an affine transformation which when applied to the eigen space of the standard PCA algorithm can duplicate the results of the face recognition algorithm.

Our strategy is superior to traditional Multidimensional Scaling (MDS) techniques since the transformation from input to output is explicitly computed. MDS just embeds the output in some space and there is no guidance in MDS methods on how to map a "new" data point onto the MDS space. However the affine approximation algorithm will be able to do the job.

The data for training and testing was taken from the FERET database. The linear discriminant analysis (LDA) face recognition algorithm and the bayesian intrapersonal and extrapersonal classifier face recognition algorithm were tested to evaluate the performance of the affine approximation algorithm. The PCA algorithm was also chosen mainly to reaffirm the affine approximation strategy.

Two training scenarios were considered - one in which both the algorithms were trained on the same set and in the other algorithm both the algorithms were trained on different sets. The closeness between the similarity matrix generated by the affine approximation algorithm and the face recognition algorithm was characterized by different techniques. Gross error measures - normalized RMS and Stress-1 error measures were used to compare the raw similarity scores. The performance in the verification and identification scenarios were compared with standard CMS and ROC curves. Finally the statistical significance of the results was tested using McNemar’s test.
As anticipated for the first training scenario the affine transformation matrix was an identity matrix when the input algorithm was the PCA algorithm. Excellent results were obtained for the LDA and Bayesian algorithm also. The results got better as the dimensionality of the affine approximation algorithm increased. In the case of PCA algorithm and LDA algorithm the results reached a plateau once the dimensionality of the affine approximation algorithm became equal to the dimensionality of the eigen space (in the case of the PCA algorithm) and the LDA space (in the case of the LDA algorithm). The bayesian algorithm showed improved results as the dimensionality of the affine approximation algorithm was increased. McNemar’s test showed that once the dimensionality became 60 the difference between the performance in identification and verification scenarios became insignificant at rank 1.

In the second training scenario, owing to the difference in the training sets, the results were not as good as the first training scenario. However a similar relationship between the errors and the dimensionality was observed. The difference between the performance in identification and verification scenarios became insignificant at rank 1 only at higher ranks.

The affine approximation algorithm would this serve as a very important benchmark for comparing face recognition algorithms more closely with PCA algorithm. The PCA algorithm has often been used as a baseline algorithm for face recognition [16, 21]. In order to allow face recognition to mature as a science better not alone is it important to establish a baseline algorithm but it is also essential to develop good techniques to compare the performance of any face recognition algorithm with the baseline algorithm.

6.1 Future Work

An important goal of the future work in this area would be to increase the size of the training set. It would be interesting to study the performance of affine approximation algorithm with more number of face recognition algorithms. Only images of type fa and fb from the FERET data set were used for training and testing. Further experimentation needs to be done by using other categories of images also.
REFERENCES


Appendix A More Results

In order to study the effect of the database another set of experiments were repeated with a different data set. As before three disjoint images sets $C_1$, $C_2$ and $C_3$ were selected from the FERET database [16, 18]. All the three sets, $C_1$, $C_2$ and $C_3$ consisted of images of the type $fa$ (regular facial expression) and $fb$ (alternate facial expression of the subject taken on the same day). Sets $C_1$ and $C_2$ consisted of 100 images of approximately 20-40 subjects. Set $C_3$ consisted of 482 images of 200 subjects. Sets $C_1$ and $C_2$ were used to train the face recognition algorithms and the affine approximation strategy while set $C_3$ was used as a validation set to compare the similarity matrix generated by the face recognition algorithm and the affine approximation algorithm. The gallery and the probe set were chosen from the validation set $C_3$. The gallery set consisted of images of type $fa$ and the probe set consisted of images of type $fb$. It was later discovered in most of the images in $C_1$ and $C_2$ the subjects were wearing spectacles. This was captured in all the eigen vectors. Figure A.1 and Figure A.2 show the top eigen faces of training set $C_1$ and training set $C_2$ respectively. The Euclidean distance measure is used for computing the distances between the gallery and the probe images in the PCA face recognition algorithm.
Appendix A (Continued)

Figure A.2. Eigen Faces of Set $C_2$.

Figure A.3. Training Scenario 1 - Plot of RMS Error. Dimensionality of the PCA step in all the three algorithms was set to 60 dimensions.
Appendix A (Continued)

Figure A.4. Training Scenario 1 - Stress Plot. Dimensionality of the PCA step in all the three algorithms was set to 60 dimensions.

Table A.1. Training Scenario 2: Average RMS Error.

<table>
<thead>
<tr>
<th>Cutoff</th>
<th>PCA</th>
<th>LDA</th>
<th>Bayesian</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>6.227</td>
<td>4.1404</td>
<td>0.0229</td>
</tr>
<tr>
<td>40</td>
<td>6.3186</td>
<td>2.8581</td>
<td>0.0344</td>
</tr>
<tr>
<td>60</td>
<td>6.4896</td>
<td>1.9132</td>
<td>0.05</td>
</tr>
<tr>
<td>80</td>
<td>6.5312</td>
<td>1.767</td>
<td>0.122</td>
</tr>
</tbody>
</table>

Table A.2. Training Scenario 2: Stress 1.

<table>
<thead>
<tr>
<th>Cutoff</th>
<th>PCA</th>
<th>LDA</th>
<th>Bayesian</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.1522</td>
<td>0.2447</td>
<td>0.2627</td>
</tr>
<tr>
<td>40</td>
<td>0.1437</td>
<td>0.248</td>
<td>0.2233</td>
</tr>
<tr>
<td>60</td>
<td>0.1414</td>
<td>0.2615</td>
<td>0.1707</td>
</tr>
<tr>
<td>80</td>
<td>0.1397</td>
<td>0.2474</td>
<td>0.167</td>
</tr>
</tbody>
</table>
Figure A.5. Training Scenario 1: CMC Curve for PCA Algorithm.
Appendix A (Continued)

Figure A.6. Training Scenario 1: ROC Curve for PCA Algorithm.
Figure A.7. Training Scenario 1: CMC Curve for LDA Algorithm.
Figure A.8. Training Scenario 1: ROC Curve for LDA Algorithm.
Figure A.9. Training Scenario 1: CMC Curve for Bayesian Algorithm.
Figure A.10. Training Scenario 1: ROC Curve for Bayesian Algorithm.
Figure A.11. Training Scenario 2: CMC Curve for PCA Algorithm.
Figure A.12. Training Scenario 2: ROC Curve for PCA Algorithm.
Figure A.13. Training Scenario 2: CMC Curve for LDA Algorithm.
Figure A.14. Training Scenario 2: ROC Curve for LDA Algorithm.
Appendix A (Continued)

Figure A.15. Training Scenario 2: CMC Curve for Bayesian Algorithm.
Figure A.16. Training Scenario 2: ROC Curve for Bayesian Algorithm.
Appendix A (Continued)

Table A.3. McNemar’s Test for Training Scenario 1 at Rank 1. SF denote the number of times the face recognition algorithm succeeds in recognizing the image but the affine approximation algorithm fails. FS denote the number of times the affine approximation algorithm succeeds in recognizing the image but the face recognition algorithm fails.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SF</th>
<th>FS</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LDA</td>
<td>3</td>
<td>0</td>
<td>0.2482</td>
</tr>
<tr>
<td>Bayesian</td>
<td>9</td>
<td>4</td>
<td>0.2673</td>
</tr>
</tbody>
</table>

Table A.4. McNemar’s Test for Training Scenario 2 at Rank 1. The rank column denotes the rank from which the difference becomes insignificant.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SF</th>
<th>FS</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>14</td>
<td>9</td>
<td>0.4042</td>
</tr>
<tr>
<td>LDA</td>
<td>13</td>
<td>6</td>
<td>0.1687</td>
</tr>
<tr>
<td>Bayesian</td>
<td>10</td>
<td>4</td>
<td>0.1814</td>
</tr>
</tbody>
</table>

Table A.5. McNemar’s Test for Training Scenario 2. The rank column denotes the rank from which the difference becomes insignificant.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rank</th>
<th>SF</th>
<th>FS</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>0.1336</td>
</tr>
<tr>
<td>LDA</td>
<td>1</td>
<td>13</td>
<td>6</td>
<td>0.1687</td>
</tr>
<tr>
<td>Bayesian</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>0.1814</td>
</tr>
</tbody>
</table>