1-1-2004

Face Recognition: Study and Comparison of PCA and EBGM Algorithms

Sachin Katadound
Western Kentucky University

FACE RECOGNITION: STUDY AND COMPARISON OF PCA AND EBGM ALGORITHMS

Date Recommended 05/05/04

Dr. Rob Byrd, Director of Thesis

Dr. Robert Crawford

Dr. James Gary

Elmer Gray, Dean of Graduate Studies and Research, (05/05/2004)

FACE RECOGNITION: STUDY AND COMPARISON OF PCA AND EBGM ALGORITHMS

A Thesis

Presented to

The Faculty of the Department of Computer Science

Western Kentucky University

Bowling Green, Kentucky

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Sachin Katakdound

May 2004

ACKNOWLEDGEMENTS

I would like to express my gratitude to all those who gave me an opportunity to learn and explore new dimensions of academic research. I am deeply grateful to my thesis Director Dr. Rob Byrd, who taught, helped, and directed me in right direction to complete the thesis paper successfully. Dr. Rob Byrd also helped me immensely in fixing grammatical errors, semantics, and reviewing this thesis paper. I would also like to thank my graduate advisors on the graduate committee, Dr. Robert Crawford, and Dr. James Gary who reviewed the thesis document.

I am indebted to Technical Agent, Patrick Grother of NIST who helped me out in implementation of experimental setup. I am obliged to Albert Lionelle, and J. Ross Beveridge of Colorado State University who facilitated in implementing Face Identification and Evaluation system provided by Colorado State University. I would like to thank my manager Masako Barnaby and Scott Copus who allowed me to have flexible schedule at work to complete thesis successfully. Finally, I take this opportunity to thank all my family and friends who showed unconditional support and confidence in me in all this time.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

In the current thesis paper we used the following conventions: a variable is represented in regular, small letter font, a matrix in capital and bold letter font, a vector in bold and italic letter font, images in capital, bold and italic letter font. Purist will argue about the difference between the terms ‘identification’ and ‘recognition’, but in the present work we use the words interchangeably to refer to face identification/recognition process.

FACE RECOGNITION: STUDY AND COMPARISON OF PCA AND EBGM ALGORITHMS

Name Sachin Katakdound Date 05/05/2004 Pages 126

Directed by: Dr. Rob Byrd

Department of Computer Science Western Kentucky University

ABSTRACT OF THESIS

Face recognition is a complex and difficult process due to various factors such as variability of illumination, occlusion, face specific characteristics like hair, glasses, beard, etc., and other similar problems affecting computer vision problems. Using a system that offers robust and consistent results for face recognition, various applications such as identification for law enforcement, secure system access, computer human interaction, etc., can be automated successfully. Different methods exist to solve the face recognition problem. Principal component analysis, Independent component analysis, and linear discriminant analysis are few other statistical techniques that are commonly used in solving the face recognition problem. Genetic algorithm, elastic bunch graph matching, artificial neural network, etc. are few of the techniques that have been proposed and implemented.

The objective of this thesis paper is to provide insight into different methods available for face recognition, and explore methods that provided an efficient and feasible solution. Factors affecting the result of face recognition and the

preprocessing steps that eliminate such abnormalities are also discussed briefly. Principal Component Analysis (PCA) is the most efficient and reliable method known for at least past eight years. Elastic bunch graph matching (EBGM) technique is one of the promising techniques that we studied in this thesis work. We also found better results with EBGM method than PCA in the current thesis paper. We recommend use of a hybrid technique involving the EBGM algorithm to obtain better results. Though, the EBGM method took a long time to train and generate distance measures for the given gallery images compared to PCA. But, we obtained better cumulative match score (CMS) results for the EBGM in comparison to the PCA method. Other promising techniques that can be explored separately in other paper include Genetic algorithm based methods, Mixture of principal components, and Gabor wavelet techniques.

1 INTRODUCTION

In this section we give an overview of the general method used to solve the face recognition problem and explore the involved steps briefly. In section 2 we discuss available face recognition methods along with other hybrid techniques. Section 2.9 summarizes the findings and recommends the techniques that are promising.

Automatic facial recognition is a complex process that can be categorized to consist of three major phases:

a. Image acquisition.
b. Image Analysis (identification of potential face regions).
c. Comparison (matching identified face regions against face database).

Image acquisition phase output is not reliable because of restricted camera resolution. Additionally results obtained from the face identification phase are hampered further because of the unconstrained environment in which the image is taken. Face recognition becomes difficult due to above stated reasons and other factors that are discussed in detail in section 2.

1.1 Image acquisition

There are various methods available to acquire an image [20], [19], [6], [18], [23]. The most common method is using digital cameras [20]. There are other methods that use different technology such as laser scan [23] and 3D color cameras [19]. Digital cameras are preferred over other image acquiring methods

due to low cost and easy availability of necessary equipment components (e.g. Charge-Coupled Device) that are necessary to capture an image. We will discuss Charge-Coupled Device (CCD) in detail here.

A CCD consists of a rectangular silicon wafer rather than the traditional photosensitive film. This silicon wafer is segmented into an array of individual light sensitive cells called “photosites”. Each photosite represents one element of the captured picture hence it is called a picture element or “pixel”.

The CCD photosites accomplish the task of sensing incoming light through the photoelectric effect, which is a characterization of the action of certain materials to release an electron when hit with a photon of light. The electrons emitted within the CCD are fenced within nonconductive boundaries, so that they remain within the area of the photon strike. As long as light is allowed to impinge on a photosite, electrons will accumulate in that pixel. When the source of light is extinguished (i.e. the shutter is closed), simple electronic circuitry and a microprocessor or computer are used to unload the CCD array, count the electrons in each pixel, and process the resulting data into an image on a video monitor or other output media [25].

As stated before a “photosite” represents a pixel and hence the size of the segmented silicon wafer array determines the smallest area of the whole picture that represents an element. This leads to two distinct characteristics that must be considered:

a. Since a CCD is an electronic component, it is sensitive to heat within the camera as well as light from the object of interest.

b. The individual photosites in the CCD array may vary significantly in their sensitivity to both heat and light.

The first factor generating a deviated result is due to heat and the electrons generated by heat. The electrons so generated by heat rather than by light need to be subtracted from the final tally of electrons in each pixel so that a true image can be rendered, this is called ***dark subtraction***. Dark subtraction is accomplished by subtracting a "dark frame" from the object image (called a "light frame"). The dark frame is created by taking an exposure in complete darkness; this exposure must be the same duration as the light frame and be made with the CCD at the same temperature as during the light frame so that electrons generated during the dark frame replicate the heat-generated electrons present in the light frame.

To remedy the second factor, the variance in electron depth across the CCD array, due to inherent differences among the pixels, needs to be leveled by dividing each pixel value by the array's average pixel value. This is called ***flat fielding***. Flat field images are made by taking a picture of an evenly illuminated scene, such as the sky at dusk or the flat gray interior of an observatory dome. The resulting image shows the inherent variances in pixel value across the CCD array due to differences in photosite sensitivity. Using mathematical algorithms implementation, all pixel values in the flat field image are divided by the array's average pixel value. After correlating pixels of the flat field image with those of the light image, a better representation of the object of interest is achieved.

Other than the two reasons stated before that might give inconsistent image representation, there are other parameters such as illumination variation, pose variation, and expression that affect the result significantly.

The digital representation of the captured image is provided as input to the next phase for further analysis (identification of potential face regions).

1.2 Image Analysis

The image analysis phase is responsible for identification of potential face candidates. In order to identify the face, the captured image needs to be pre-processed to “nullify” the extreme variations in the image that generate inconsistent results and represent the important features that are presented as input to the next phase.

Image analysis can be partitioned into preprocessing, feature extraction, and representation.

1.2.1 Preprocessing

Preprocessing removes the change in face appearance caused by the pose variations and illumination changes. We will discuss here briefly pose and illumination variations.

Pose variations are caused by changes in the angle and distance at which a given face is being observed. Thus, pose variations occur due to scale changes as well as in-plane and out-of-plane rotations of faces. Out-of-plane changes are difficult to handle. In fact, out-of-plane rotations can be addressed only to a certain extent by warping techniques, where the center positions of distinctive facial features are used as reference points to normalize test faces according to

some generic face models. On the other hand, scale changes can be handled by using a normalization process based on a stretching transform.

Illumination variations can be almost eliminated by filtering the input image with Gabor wavelets [14], [16], [10]. Removal of variation due to a partly lightened face is difficult. Specular reflections on eyes, teeth and wet skin may be corrected using brightness models.

The preprocessing step applies face normalization (or use of normalized extracted features) that generates scale, illumination, and pose invariant data. ***Face segmentation*** is applied next to segregate face regions from other background regions in the image. Face segmentation allows isolation of transient and intransient features (discussed in section 2.b) within faces or can be used to separate faces of interest from the background as explained before.

1.2.2 Feature extraction phase

Feature extraction phase identifies and locates important face features in the given image and quantitatively represents the distinctive face characteristics without general loss of information. In short, this phase generates a compressed representation of face features, and, after eliminating extraneous data only critical face characteristics remain.

B. Fasel and Juergen Luettin discuss feature extraction [17] techniques to extract crucial facial features to represent facial expressions. It is possible to achieve a pose, illumination, and facial expression invariant system, by considering the techniques used by B. Fasel et al. in our face recognition process.

Facial features can be distinguished into two types. *Intransient facial features* are the features always present in the face, but may be deformed due to facial expressions; among these are the eyes, eyebrows, the mouth, tissue texture, facial hair as well as permanent furrows that influence the appearance of facial expressions. *Transient facial features* cover different kind of wrinkles and bulges that occur with facial expressions. The forefront and the regions surrounding the mouth and the eyes are prone to contain transient facial features.

Feature Extraction methods can be categorized as either deformation extraction or motion extraction models. In addition, each extraction model can be further subdivided based on either the holistic or local approach. The whole face image is used for feature extraction in holistic approach, whereas the local approach is based only on identified face characteristics such as eyes, eyebrows, nose, and mouth.

1.2.2.1 Deformation extraction

Deformation based techniques rely on neutral face images or face models to extract facial features that are relevant to facial recognition. This technique can be applied to both single images and sequences of images independent of each other. Deformation of facial features is characterized by shape and texture changes and leads to high spatial gradients that are good indicators for facial actions. These may be analyzed either in the image or in the spatial frequency domain.

Deformation extraction can be further classified as *image-based* or *model-based* deformation extraction.

In ***image-based*** techniques, features are extracted without prior knowledge about the object of interest, and are typically fast and simple [11], [1], [7], [14]. In the presence of extensive pose variations, however, these techniques can be unreliable and unmanageable.

In ***model-based*** techniques, facial features are represented as 2D models or 3D models. With 2D models, the recognition system tries to represent facial features without attempting to recover the volumetric geometry of the scene [14], [16], whereas in 3D face models the features are represented in 3D that gives more accurate representation of facial features. Due to complex mapping procedures, 3D models are computationally demanding [19], [23].
We discuss the categorization below.

1.2.2.1.1 Holistic image-based deformation

Gabor wavelet filters are applied to whole faces prior to feature extraction [8], [14], [16], and [10]. The important condition for this model is the distinctive separation of face and background to prevent disturbance caused by cluttering. Jun Zhang, Yong Yan, and Martin Lades [14] discussed a similar technique using neural nets.

1.2.2.1.2 Local image-based deformation

In this model, important face characteristics are given more weight over other information that is subsequently used for feature extraction. Bruce Draper

et al. [1] discussed some of the commonly used techniques such as Principle component analysis (PCA) and Independent component analysis (ICA). Other papers that discussed related techniques [7], [3], [15] and [6] used a modified method employing local facial characteristics to extract important feature information.

1.2.2.1.3 Holistic model-based deformation

Model-based approaches can be distinguished as active appearance model, point distribution model, and labeled graphs.

Appearance-based model approaches allow separation of different information sources such as facial information and deformation changes [8], [23] and [22]. Lixin Fan et al. [2] used combined feature texture similarity measures as the criteria to identify and compare two different faces. L. Torres et al. [18] used a similar technique to eliminate texture variation and to analyze the importance of color information in face recognition. A major drawback of the appearance-based model is the manual labor necessary for the construction of the shape models.

Active shape models (ASM) allow simultaneous determination of shape, scale, and pose by fitting an appropriate *point distribution model* (PDM) to the object of interest. The point distribution model is based on landmark points that are precisely placed around intransient facial features during the training of the models.

Labeled graphs represent another holistic model based technique that consists of sparsely distributed fiducial feature points. The nodes of these

feature graphs consist of Gabor jets, where each jet has components representing a filter response of a specific Gabor wavelet extracted at a given point.

1.2.2.2 Motion extraction

Motion extraction approaches focus directly on facial changes occurring due to facial expressions. We can further distinguish motion extraction as *holistic dense optical flow* approaches and *local dense optical flow* [23], [22]. Models based on motion extraction are basically advanced methods of searching a given image in the video sequence. Motion-based models use typical image-based techniques combined with other temporal information of the video sequence to identify or recognize the faces. Some of the techniques based on PCA, ICA, and 3D morphable models are discussed in [24], [11], [12], [19], [10], [22], and [23].

1.2.3 Representation

Depending upon the system used to extract features, the representation of facial information differs. PCA uses eigenvectors to represent the compressed facial information as described in [1], [3], [4], [5] and [6]. Gabor wavelet techniques use fitting parameters to represent the matching with the image model. Three-dimensional modeling techniques use shape and texture representation to represent extracted facial features [22] and [23]. Most techniques represent the extracted information as a vector that captures the facial features quantitatively. Representation technique determines the computational overhead associated with the system to find the best match from the database. Systems that use extensive data to represent extracted features will

have to spend more time in comparing the extracted features between probe and gallery image. Representation techniques that capture irrelevant data also waste space because of low compression ratios.

1.3 Comparison

Once the important features are extracted and represented appropriately, the best match for the given probe representation can be obtained by comparing the probe representation against the database. Depending upon the feature representation model and the system used to solve the face recognition problem, the system uses a parameterized measure to obtain the best match.

The distance measure is the most common measure used to find the closest match. There are different ways to represent the distance measure such as Euclidean distance, city-block, cosine angle, etc. and will be discussed in section 2. On the other hand, 3D models use deformation coefficient parameters that cause the given image to fit the generic model after deformation. This deformation coefficient is the comparison measure for 3D models. Using appropriate parametric representation and the specific comparison criteria, the system can obtain the best match.

2 BACKGROUND

Most commercial face recognition systems use two main steps to achieve face recognition: Subspace Projection and Classification in the compressed

space. Face recognition applications include identification for law enforcement, authentication for banking and security system access, human-computer interaction, and customizable animation.

Ideally, the face recognition technique must be invariant to internal and external environment factors that hamper the face recognition process. Internal characteristics that affect face recognition can be identified as pose, face texture, and expression variations (intransient and transient facial features variations). External characteristics produce unreliable results due to uncontrolled environment under which the image is obtained. The uncontrolled environment factors include varying illumination, occlusion, and similar external factors. A few of the face recognition techniques are Artificial Neural Networks (ANN), PCA, ICA, Gabor jets, Linear Discriminant Analysis (LDA or Fisher-faces), and Hybrid methods.

2.1 Template Matching Technique

Template matching is a simple technique in which a template is used as the criteria to identify the face region in the given scanned image. There are different types of templates that help determine important characteristics of the human face. Alper Yilmaz and Mubarak Shah [11] used a modified template matching technique in determining potential face candidates in a video sequence. In another instance, Markus et al. [12] used a deformable template matching technique to estimate the position of the eye features with a simplified cost function. After an approximate estimate of the eye features, the image can be analyzed further to identify other important facial characteristics such as

nose, eyebrows, and mouth. Face recognition can be achieved using information obtained from the identified features.

This technique is relatively simple and easy to implement, but lacks consistent results because of sensitivity to various environment factors such as shape, pose, and illumination variation.

2.2 Statistical Techniques

Statistical techniques rely on statistical analysis of facial features. A few of the statistical techniques that are used in face recognition are PCA, ICA and LDA.

2.2.1 Principal Component Analysis (PCA)

PCA is an unsupervised technique that optimally compresses the facial information so that the mean squared error between the original images, and for the given level of compression the reconstruction, is minimized [1], [4], [5], [7], [3], and [6]. Turk and Pentland used PCA extensively for face recognition. They used PCA to compute a set of subspace basis vectors for a database of face images. These basis vectors, when represented as an image, correspond to face-like structures called *eigenfaces*. Image projection in this compressed subspace allows for easy comparison of images against the set of images from the database.

PCA basis vectors are obtained from the set of training images $\boldsymbol{I}$. In simple terms the average image in $\boldsymbol{I}$ is computed and this mean image of $\boldsymbol{I}$ is subtracted from the training images to obtain a set of data samples. These data samples are arranged in a matrix $\mathbf{X}$, with one column of $\mathbf{X}$ representing a sample image.

$\mathbf{XX}^T$ represents the sample covariance matrix for the training images. The principal components of the covariance matrix are computed from the relationship, $\mathbf{R}^T(\mathbf{XX}^T)\mathbf{R} = \mathbf{D}$, where **D** is the diagonal matrix of eigenvalues and **R** is the matrix of orthonormal eigenvectors.

The size of the eigenvectors generated is dependent on the size of the training image. So, to reduce the complexity and computational overhead, a simple approach of reducing eigenvectors is used: only the *n* largest eigenvectors associated with the largest eigenvalues are used to define the subspace, where *n* is the desired subspace dimensionality.

In geometric representation, **R** is a rotation matrix that rotates the original coordinate system onto the eigenvectors. The eigenvector is associated with the largest eigenvalue, representing the axis of maximum variance. The eigenvector associated with the second largest eigenvalue is the orthogonal axis with the second largest variance, etc.

PCA, being a statistical technique, has additional data constraints. Compression of facial information allows significant representation of meaningful data in reduced dimensions. This helps in reducing the computational overhead time.

A normal distribution must be satisfied to get accurate probabilistic representation of important eigenvectors from training samples that attribute to reliable face recognition. It is assumed that the axes of large variance probably correspond to the signal, while axes of small variance are probably noise.

Finally, normalized images must be used to make facial images independent of the variation caused by illumination, texture and other unconstrained environment features. By subtracting the mean image from the training images, scaling images to form unit vectors, and assuming that training set consists of Gaussian normal distribution data, we can use Euclidean distance to represent the correlation between the source images.

The PCA technique can be formulated in following steps:

a. Represent every face image as a vector and express it in an orthogonal basis, computed from the training images set.

b. Estimate the covariance matrix **C** for *n* training vectors $\boldsymbol{x}_1$', $\boldsymbol{x}_2$',...$\boldsymbol{x}_n$' as,

$$\mathbf{C} = \frac{1}{n\left(\Sigma_k \left(\boldsymbol{x}_k' - \mu'\right)\left(\boldsymbol{x}_k' - \mu'\right)^{\mathbf{T}}\right)}$$

Where *k* varies over *n* and

μ = the estimation of the training vectors expectation.

c. Compute the *n-1* eigenvectors $\boldsymbol{e}_0$',... $\boldsymbol{e}_{n-1}$', also known as eigenfaces, from this covariance matrix and sort them by descending value of their eigenvalues $\lambda_1 >= \lambda_2 >= \ldots >= \lambda_n$.

d. Represent the image by projecting the image into this eigenspace, using a scalar product. Any training image can be obtained by a linear combination of the eigenfaces:

$$\boldsymbol{x}' = \mu' + \Sigma_i^{n-1} \left(\boldsymbol{x}_i'' \; \boldsymbol{e}_i'\right)$$

where

$$\boldsymbol{x}_i'' = \boldsymbol{x}_i' \cdot \boldsymbol{e}_i'$$

e. Express a test vector $\boldsymbol{y'}$ in the eigenspace as,

$$\boldsymbol{y'} = \mu' + \Sigma_i^{n-1} (\boldsymbol{x}_i^{''} \boldsymbol{e}_i^{'})$$

where

$$\boldsymbol{x}_i^{''} = \boldsymbol{x}_i^{'} \boldsymbol{.} \boldsymbol{e}_i^{'}$$

The training and test vectors are then compared using a distance measure. The selected training image is the one that has the minimal distance under the eigenspace.

The distance measures that can be used are discussed in section 2.2.4. The eigenface concept can be extended further to use important features of the face such as the eyes, the nose, and the mouth in reliable determination of a face against the face database.

L. Torres et al. used the concept of PCA in determining the face in a video sequence [24]. They used a set of eigenfaces that were to be searched in the given sequence. This technique is called self-eigenface approach.

2.2.2 Independent Component Analysis (ICA)

Unlike PCA, ICA minimizes both second-order and higher-order dependencies in the input. It is closely based on the blind source separation (BSS) problem [1] and [6], where the goal is to decompose an observed signal into a linear combination of undetermined independent signals.

Let $\boldsymbol{s}$ be the vector of undetermined source signals. The mixing model is given as:

$$\boldsymbol{x} = \mathbf{A}\boldsymbol{s}$$

where **A** is the undetermined mixing matrix, and ***x*** is the vector of observed mixtures. In order to obtain the matrix **A**, the observed mixture can be processed to separate the matrix A from the actual source signal **s**. This model is represented below.

$$\boldsymbol{u} = \mathbf{W}\boldsymbol{x} = \mathbf{WA}\boldsymbol{s},$$

where **W** is separating matrix. With the assumption that the source signals are independent of each other and the mixing matrix **A** is invertible, the ICA algorithm tries to obtain the mixing matrix **A**, or the separating matrix **W**.

Whiteness is a stronger constraint that requires both de-correlation and unit variance. The whitening transform can be determined by $\mathbf{D}^{-1/2}\,\mathbf{R}^{T}$ where **D** is the diagonal matrix of the eigenvalues and **R** is the matrix of orthogonal eigenvectors of the sample covariance matrix. Applying only whitening to observed mixtures provides the source signal only up to an orthogonal transformation, but ICA goes much further in transforming the whitened data into a set of independent signals.

A signal is termed statistically independent when $\boldsymbol{f}_u(u) = \Pi_i\, \boldsymbol{f}_{ui}(u_i)$**,** where $\boldsymbol{f}_{\boldsymbol{u}}$ is the probability density function of *u*. Since **W** cannot be calculated using any closed form expression, the approximate **W** is obtained by using numerical analysis. This approximation process is also called learning process and weight matrix **W** is also known as learned weight matrix.

Instead of maximizing the independence condition above directly, most ICA algorithms recast the problem as iteratively optimizing a smooth function whose global optima occur when the output vectors ***u*** are independent.

Some of the most commonly used algorithms are discussed below.

Infomax, is an algorithm that uses the entropy maximization as an indirect indicator to obtain the maximum independence. Infomax performs gradient ascent on the elements w_{ij} so as to maximize H(u).

$$\mathbf{H(u) = -} \int_{\mathbf{u}} \boldsymbol{f}_u(\boldsymbol{u}) \log \boldsymbol{f}_u(\boldsymbol{u}_i)\mathrm{du}$$

The *JADE* algorithm minimizes the kurtosis of $\boldsymbol{f}_u(u)$ through a joint diagonalization of the fourth order cumulants. Minimization of kurtosis maximizes the statistical independence.

FastICA uses the following relation, $\mathbf{J(}\boldsymbol{y}\mathbf{) = c[E\{G(}\boldsymbol{y}\mathbf{)\}-E\{G(}\boldsymbol{v}\mathbf{)\}]^2}$, where $\boldsymbol{G}$ is a non-quadratic function; v is Gaussian random variable, and **c** is any positive constant. It can be shown that maximizing any function of this form will also maximize the independence. Of the discussed algorithms methods FastICA is the most commonly used algorithm.

ICA can also be categorized on the basis of architecture used in [1].

2.2.2.1 Architecture I: Statistically Independent Basis Images

In this architecture, the input face images in **X** are considered to be a linear mixture of statistically independent basis images $\boldsymbol{S}$ combined by an unknown mixing matrix **A**. The face images are variables and the pixel values provide observations for the variables. The ICA algorithm learns the weight of matrix **W**, which is used to recover a set of independent basis images in the rows of $\boldsymbol{U}$. Projecting the input images onto the learned weight vectors produces the independent basis images. The compressed representation of a face image is a

vector of coefficients used to generate the image by linearly combining the independent basis images.

Use of PCA as a pre-processor in a two-step process allows ICA to create subspaces of size *m* for any *n*. It is also argued that pre-applying PCA enhances ICA performance by discarding small trailing eigenvalues before whitening, and reducing the computational complexity by minimizing pair-wise dependencies.

Mathematically, the *m* independent basis images in the rows of ***U*** are computed as $\mathbf{U = W * R^T}$, where **R** is a *p* by *m* matrix containing the first *m* eigenvectors of a set of *n* face images.

Let *p* be the number of pixels in a training image. The rows of the input matrix to ICA are variables and the columns are observations. ICA is then performed on $\mathbf{R^T}$.

The *n* by *m* ICA coefficients matrix **B** for the linear combination of independent basis images in ***U*** is computed as follows: Let **C** be the *n* by *m* matrix of PCA coefficients. Then,

$$\mathbf{C = X * R} \text{ and } \mathbf{X = C * R^T}$$

Since $\mathbf{U = W * R^T}$ and assuming that **W** is invertible,

$$\mathbf{R^T = W^{-1} * U}$$

$$\mathbf{X = C * R^T}$$

$$\mathbf{X = C * W^{-1} * U}$$

$$\mathbf{X = (C*W^{-1}) * U = B * U}$$

Each row of **B** contains the coefficients for linearly combining the basis images to comprise the face image in the corresponding row of **X**. **X** is the reconstruction of the original data with the minimum squared error as in PCA.

2.2.2.2 Architecture II: Statistically Independent Coefficients

In statistically independent coefficient architecture, the system tries to achieve statistical independent coefficients for the input data. The input is transposed in comparison to architecture I. The pixels are variables, and the images are observations. The source separation is performed on the pixels, and each row of the learned weight matrix **W** is an image.

The inverse matrix of **W** contains the basis images in its columns. The statistically independent source coefficients in S that comprise the input images are recovered in the columns of ***U***. The statistically independent coefficients are computed as $\mathbf{U = W * C}^{T}$. Note that the **R*A** columns represent the actual basis images.

A comprehensive analysis by B. Draper et al. [1] indicated that the performance of ICA technique depends upon the task for which the technique is used.

The choice of architecture is decided by the task at hand. For architecture I, since images are used as variables and pixels are taken as observations, it is more suitable for facial action or facial expression analysis. On the other hand, ICA architecture II uses pixels as the variables and images as the observations, therefore this technique is more suitable for the face recognition task.

Kunio Takaya and Kyung-Yunh Choi, proposed the use of ICA technique in the detection of facial components in video sequences [6]. We will explain briefly the modified ICA technique used to detect faces in the video sequence.

The primary goal [6] is mapping the detected facial components such as eyes and mouth to a 3D wire frame model. Two approaches are explored the face Localization Technique, and the wavelet Sub band filter

In order to develop a 3D facial model from video images that are 2D, we can apply artificial intelligence techniques aided by a database that supplies any missing information to construct an appropriate 3D wire frame facial model.

Generating a 3D wire frame from a video sequence can be deemed as a two-step process.

1. Identify and localize the face in the video sequence.
2. Extract facial features that develop 3D wire frame model from the extracted features.

Before developing a 3D model we first identify the face in the video sequence. This can be achieved by simple skin color filtering on an individual frame, and using the human face characteristics knowledge. After a face is identified and localized, we then extract facial features that are used to synthesize a face model and to track its motion.

Independent Component Analysis is a powerful method that retains the higher order statistical information without losing pertinent information. Principal Component Analysis (PCA) transfers the data from a higher dimension to a smaller subspace. This is achieved by reducing the dimensionality, but this

may lose some vital information. ICA works well in extracting facial components only when the human face is localized. The same results will be hard to achieve without localizing the face, due to the relative small size of facial features such as the eyes, and mouth with respect to other facial components and background information.

2.2.3 Linear Discriminant Analysis (LDA)

LDA is a supervised learning technique that relies on class labels. For the statistical techniques like PCA and LDA the resulting feature vectors are spatially placed global vectors [1]. This indicates that a change to a single input pixel will alter every dimension of its subspace projection.

Multidimensional space provides us with additional information for classification. However, the extremely large amount of data involved makes it equally difficult for the system to find the most appropriate hyperplane for classification. The conventional way to handle this problem is to preprocess the data so as to reduce its dimensionality before applying a classification algorithm. Fisher's Linear Discriminant uses a linear projection of the n-dimensional data onto a one-dimensional space (i.e., a line) [26]. With the expectation that the projection onto a line will be well separated by class, the classification problem becomes choosing a line that is so oriented to maximize class separation without causing the data to be intermingled.

Consider an input vector $\boldsymbol{x}$ that is projected on a line as a scalar value y, and is given by, $y = \boldsymbol{w}^{T}\boldsymbol{x}$, where $\boldsymbol{w}$ is a vector of an adjustable weight parameter. By

adjusting the components of the weight vector ***w***, we can select a projection that maximizes the class separation.

J. Ross et al. [7] explored the combination of LDA with PCA as a preprocessing step. Fisher's method defines *c*-1 basis vectors where *c* is the number of classes. This means that number of classes must be known before hand. That is very difficult in problems such as face recognition. In [7] the basis vectors were expressed as rows in a matrix **W**, and the discriminants were defined as those basis vectors that maximized the ratio of distance between classes divided by distances within each class:

$$J(\mathbf{W}) = \frac{(\mathbf{W}\ \mathbf{M}_B\ \mathbf{W}^T)}{(\mathbf{W}\ \mathbf{M}_w\ \mathbf{W}^T)}, \text{ where}$$

$$\mathbf{M}_w = \Sigma_i^c\ \mathbf{M}_i, \text{ and}$$

$$\mathbf{M}_i = \Sigma_j^{ni}\ (\mathbf{y}_j - \mu_i)\ (\mathbf{y}_j - \mu_i)^T, \text{ and}$$

$$M_B = \Sigma_j^{ni}\ (\mu_i - \mu)\ (\mu_i - \mu)^T,$$

The basis vectors were the row vectors in **W** that maximized *J*(**W**). Projecting an image **y** into LDA subspace yields **y**'':

$$\mathbf{y''} = \mathbf{W}\mathbf{y'} = \mathbf{W}\ \mathbf{E_d}\ (\mathbf{y} - \mathbf{X}_\mu),$$

where $\mathbf{E_d}$ is resulting *n* by *d* orthogonal projection matrix from a PCA processing, and x_μ is the centroid of the training images, and **X** is an *n* x *m* data matrix containing each image as a column vector.

Results obtained by J. Ross [7] suggest that LDA performs uniformly worse on the FERET database compared to the PCA algorithm.

2.2.4 Comparison of Distance Measures

A typical face recognition system is based on the distance measure between the probe image and the images stored in the database as a decision parameter to determine the best match. Hence, there has been thorough analysis done on the choice of distance measures [1], [3], and [7]. In fact, for the training of the images, an optimal distance measure is needed to categorize the images and to determine the difference in the images as a function of this distance measure. Hence, distance measure is a very important part of the face recognition system.

We explain few of the most commonly used distance measures below:

2.2.4.1 City Block (L1)

City block distance (L1) is the difference between two vectors of *n* length each. For the two given vectors ***x*** and ***y***, we obtain the difference between two corresponding pair values from the two eigenvectors. Summation of difference between all pairs such as (x_1, y_1), (x_2, y_2) and so on is city block distance. Mathematically it is given as,

$$d_{Cityblock}(\boldsymbol{x}, \boldsymbol{y}) = |\boldsymbol{x} - \boldsymbol{y}| = \Sigma_{i=1}^{n} |x_i - y_i|$$

2.2.4.2 Euclidean Distance (L2)

The Euclidean distance is the straight-line distance between any two points *a* and *b* in the same plane. For point *a* at (x_1, y_1) and point *b* at (x_2, y_2), L2 distance is given as, $\sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)}$. For any two vectors ***x*** and ***y*** of length *n* each, Euclidean distance is given as,

$$d_{Euclidean}(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||^2 = \sqrt{(\Sigma_{i=1}^{n} (x_i - y_i)^2)}$$

2.2.4.3 Cosine distance

For any two given vectors ***x*** and ***y***, the cosine distance between two given vectors is obtained by dividing the dot product of two given vectors by the product of the scalar value of the two vectors. Mathematically it is given as,

$$d_{cosine}(\boldsymbol{x}, \boldsymbol{y}) = - \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}||\ ||\boldsymbol{y}||}$$

$$= - \frac{\Sigma_i\ (x_i y_i)}{\sqrt{(\Sigma_i\ (x_i)^2\ \Sigma_i\ (y_i)^2)}}$$

where i=1..n

2.2.4.4 Yambor distance

For any two given vectors ***x*** and ***y*** of length *n*, we first multiply the scalar values of corresponding elements in the vectors and divide the product by the eigenvalue at that element location. Yambor distance is obtained by adding together all such ratios and is given as,

$$d_{yambor}(\boldsymbol{x}, \boldsymbol{y}) = - \Sigma_{i-1}^{n} \frac{x_i\ y_i}{\sqrt{(\lambda_i)}}$$

where λ_i = i^{th} eigenvalue.

This distance was initially referred to as Mahalanobis distance in the [1] and [2] papers.

2.2.4.5 Mahalanobis cosine distance

Mahalanobis cosine is the cosine of the angle between the images after they have been projected into the recognition space and have been further normalized by the variance estimates. So, for images ***U*** and ***V*** with corresponding projections ***m*** and ***n*** in Mahalanobis space, the Mahalanobis Cosine is:

$$S_{mahCosine}(\boldsymbol{U}, \boldsymbol{V}) = cos(\theta_{mn})$$

$$= \frac{|m|\,|n|\,cos(\theta_{mn})}{|m|\,|n|}$$

$$= \frac{m.n}{|m|\,|n|}$$

$$D_{mahCosine}(\boldsymbol{U}, \boldsymbol{V}) = -S_{mahCosine}(\boldsymbol{U}, \boldsymbol{V})$$

2.2.4.6 Mahalanobis L1

This measure is exactly the same as the City block measure only the distances are scaled to Mahalanobis space. So for images ***U*** and ***V*** with corresponding projections ***m*** and ***n*** in Mahalanobis space, Mahalanobis L1 is:

$$D_{mahL1}(\boldsymbol{U}, \boldsymbol{V}) = \sum_i |\boldsymbol{m}_i - \boldsymbol{n}_i|$$

2.2.4.7 Mahalanobis L2

This measure is exactly like Euclidean distance only computed in Mahalanobis space, so for images *u* and *v* with corresponding projections ***m*** and ***n*** in Mahalanobis space, Mahalanobis L2 is:

$$D_{mahL2}(\boldsymbol{U},\boldsymbol{V}) = \sqrt{(\sum_i (\boldsymbol{m}_i - \boldsymbol{n}_i)^2)}$$

This is also known commonly as the Mahalanobis distance.

2.2.4.8 Choice of Distance Measure

The simplest mechanism devised to combine more than one distance measure is to add the distances, i.e.

$$S(a_1, a_2, .., a_n) = a_1 + a_2 + \ldots + a_n,$$

where S = the function generating all combinations of $a_1,..a_n$.

Another method, Weighted Summation, combines multiple estimators independently and then combines those results by voting as proposed by J. Ross Beveridge et al in [3]. Every distance measure is allowed to vote for the image that it believes is the closest match to a probe. The image with the most votes is chosen as the matching gallery image. Voting was used in [3] and can be accomplished in three possible ways.

a) Bagging: Each classifier is given one vote.
b) Bagging, Best of five: Each classifier votes for the five gallery images that most closely match the probe image.
c) Bagging, Weighted: Each classifier casts five votes for the closest gallery image and so on casting just one vote for the fifth closest image.

To reduce the complexity and to increase the computational efficiency of the system only those eigenvectors are used that have maximum information content, i.e., all the eigenvectors that have eigenvalues significantly small compared to the rest of the eigenvalues are ignored.

2.2.4.9 Choice to Reduce Eigenvectors

There are three different techniques commonly used to reduce the eigenvectors:

1. ***Heuristic threshold***

First we order the eigenvectors based on the eigenvalue, the highest eigenvalue vector first followed by the second highest eigenvalue vector and so on. In the heuristic method we drop the last 40 percent of the eigenvectors.

2. ***Energy threshold***

The minimum number of eigenvectors to guarantee that energy *e* is greater than a threshold t is selected. The typical threshold is 0.9.

$$\boldsymbol{e}_i = \frac{\text{Sum of all Eigenvalues} <= t}{\text{Sum of all eigenvalues}}$$

3. ***The Stretching dimension***

The stretch value S_i for the i^{th} eigenvector is the ratio of that eigenvalue over the largest eigenvalue, $S_i = \lambda_i/\lambda_1$. All eigenvectors with S_i greater than the threshold value (typically 0.01) are retained. A like-image difference ω_i for each of the *m* eigenvectors is defined as, $\omega_i = \delta_i/\lambda_i$ where

$$d = S \mid x_j - y_j \mid,$$

where $j = 1..m$ and $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the images of the same person such that $\boldsymbol{X}$ is the gallery image and $\boldsymbol{Y}$ is the corresponding persons image from the probe set ordered such that $x_j \varepsilon \boldsymbol{X}$ and $y_j \varepsilon \boldsymbol{Y}$.

When the difference between the images that ought to match is large relative to the variance for that dimension λ_i, then ω_i is large. On the other hand, when the difference between the images that ought to match is small relative to the variance, ω_i is small. For each probe/gallery image match from the FERET data set, the set is reorganized on the basis of ω_i and the eigenvectors are arranged accordingly.

We studied the available distance measures and different eigenvector reduction methods in this section. But, particular choice of the distance measure that gives best recognition result is dependent on the system used to solve the face recognition problem [3], [7], and [1]. Overall, some general conclusions that we can draw from our study are given below.

2.2.4.10 Study results for the choice of the distance measure and the eigenvector selection

1. Reordering by the like-image difference improves performance for small numbers of eigenvectors, but, overall, there is no significant improvement when eigenvectors are sorted on the basis of like-image distance measures or simple distance measures.
2. Recognition rates are significantly higher for the larger eigenspace.

3. Better results are obtained using the Mahalanobis distance with a large number of eigenvectors than with a small number of eigenvectors.
4. A standard PCA classifier gives better results using the Mahalanobis distance rather than the L1, L2, or the Angle.
5. The correlation between L1, L2, Angle, and Mahalanobis distance measures with a shared bias indicate that by combining the L1 measure with other measures gives a minor improvement, overall. Combination of the distance measures does not give significant performance improvement.

2.2.5 Comparison of Statistical Techniques

The most commonly used PCA method for face recognition has been scrutinized by various authors in comparison with other similar techniques [1], [3], [4], [5], [6], and [7]. There were conflicting claims made by various authors. Barlett et al [34], Liu et al [35], and Kunio Takanya et al [6], claimed in their work that ICA is better than PCA. While, Baek et al [5] claimed that PCA is better than ICA. There were papers [36] that claimed PCA and ICA algorithms are not significantly different. But one common claim in all the papers was that the LDA technique is less efficient than the other two techniques. One reason for this is the requirement to have prior knowledge of the classes.

A comprehensive comparison was accomplished by B. Draper, K. Baek, M. Barlett, and J. Ross [1], which explored the different types of algorithms and the specific architectures in ICA and PCA. The conflicting claims were mainly because of the comparison of various algorithms in ICA against PCA

irrespective of the architecture ICA used (refer to 2.2.2 for ICA architecture). B. Draper et al. claimed that the choice of architecture determines the performance of ICA. For a holistic face recognition technique, ICA architecture I is more suitable. On the other hand, for local analysis such as face expression analysis, architecture II is more appropriate, irrespective of the algorithm used. Various algorithms in ICA were also compared against the PCA technique. The conclusions that we draw from our study are: 1. Methods giving the best results for face recognition were ICA architecture II with FastICA algorithm, and PCA with L1 or PCA with Mahalanobis distance measure. 2. For facial expression or facial action determination, the methods that give the best results in the decreasing performance were ICA architecture I using the InfoMax algorithm, and PCA with L2 distance measure.

B. Draper et al. [5] discussed the results obtained by first applying PCA to project the data into an *n* dimension subspace, and then applying InfoMax to the eigenvectors to minimize the statistical dependence among the rows of **U** in $\mathbf{W}\mathbf{R}_n^T = \mathbf{U}$. There have been claims made that using PCA as a preprocessing step enhances the results obtained with ICA. Unfortunately, even with this heuristic, the ICA basis vectors are much more expensive to compute than the PCA basis vectors. There are other similar claims made in [7], [3], and [5] that support the claim made by B. Draper et al.

A more comprehensive comparison between the statistical technique and other techniques is discussed later in section 3.

2.3 Artificial Neural Network (ANN)

ANN face recognition techniques can be categorized as subsets of pattern recognition. To classify a pattern into a category is a learning process. A pattern recognition system learns through the iterative adjustment of its synaptic weights and/or other system parameters. The assumption in performing iterations is that by repetition in the learning process, the system will become a more knowledgeable and effective system, and will produce a higher recognition rate. Learning processes are distinguished as supervised learning or unsupervised learning. The ease of constructing a linear/ nonlinear decision boundary between the different classes in a nonparametric fashion is the main advantage of using the supervised learning system to perform pattern recognition. In certain instances we do not have prior knowledge of the categories into which the patterns are to be classified. In such unsupervised learning techniques (also called clustering), patterns are associated into clusters based on certain pre-decided common properties. Real neural networks are networks of living nerve cells. Such networks of neurons possess the capability of thinking, feeling, learning, and remembering. Artificial neural networks and neuro-computers are models inspired by these brain functions [26].

A back propagating neural network can be trained with images to learn the recognition process. The network depends heavily on the number of input, output, and hidden layers. For a moderate size image the number of inputs of the network would be large hence, the complexity of the network would also be high [14]. To reduce complexity [30], the network can be divided into two back

propagation networks, the auto association network, and the classification network.

The auto association network has n inputs, n outputs and p hidden layer nodes, such that p is significantly less than n. For the given input face vector $\mathbf{x}$ the network produces y that is the closest representation of x. The hidden layer output $\boldsymbol{h}$ provides a compressed version of $\mathbf{x}$, or a feature vector. The output from the auto association network is given as input to the classification network.

For a linear network the feature vector is equivalent to the feature vector generated by eigenfaces. For a non-linear the network feature vector may not be the best representation of information.

For each training face vector $\mathbf{x}_k$, k = 1..N, the output from the auto association network can be represented as h_k, where h_k is p dimensional such that $p << n$ and $p <$ N and the output y_k such that

$$h_k = F(\mathbf{W}_1 \boldsymbol{x}_k),\ \boldsymbol{y}_k = \mathbf{W}_2\ h_k$$

$\mathbf{W}_1$ (p x n) and $\mathbf{W}_2$ (n x p) are corresponding weight matrices and $\boldsymbol{F}(.)$ can be a linear or non-linear function, applied “component-by-component”. For a matrix consisting of x_k, y_k, and h_k we can rewrite the above formula as,

$$\mathbf{H} = F(\mathbf{W}_1\mathbf{X}),$$

$$\mathbf{Y} = \mathbf{W}_2\mathbf{H}$$

The training error can be minimized using the Frobenius matrix norm,

$$||\mathbf{X}\text{-}\mathbf{Y}||^2 = \Sigma_k^{\,n}\ ||x_k - y_k||^2$$

For $\mathbf{Y} = \mathbf{W}_2\mathbf{H}$, the rank is no more than p, and is the best rank-p approximation to $\mathbf{X}$, hence,

$$\mathbf{W}_2\mathbf{H} = \mathbf{U}_p \triangle \mathbf{V}_p^T$$

where $\mathbf{U}_p = [u_1, u_2, \ldots, u_P]^T$, $\mathbf{V}_p = [v_1, v_2, \ldots, v_P]^T$ are the first p left and right singular vectors in the SVD of $\mathbf{X}$ and also are the first p eigenvectors of $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$. To achieve this optimum, F(.) must be a linear function and the weights must be set as,

$$\mathbf{W}_1^T = \mathbf{W}_2 = \mathbf{U}_p$$

For any input x,

$$\mathbf{h} = \mathbf{W}_1 x = \mathbf{U}_p\mathbf{x} = [\langle \mathbf{u}_1, x\rangle, \langle \mathbf{u}_2, x\rangle, \ldots., \langle \mathbf{u}_p, x\rangle]^T = \mathbf{X'}$$

and represents the feature vector in the eigenface approach. An auto association network trained using the back propagation algorithm with a non-linear function cannot achieve the optimal performance achieved with the help of a linear function.

2.4 Genetic Algorithm (GA)

The recognition of a face can be formulated as a pattern recognition problem that subsequently can be formulated further as a learning problem. Angel Fernando and Kurl-Morales used Genetic Algorithms along with Multivariate analysis [15] in exploring the solution for face recognition. The method relies on supervised learning, consisting of the acquisition of the classification functions from a set of examples or training images. The goal of a learning function can be formalized as capturing the “general pattern” in the training data. To represent a face with pixel attributes obtained from the image, we use a

family of approximating polynomials. By finding such an approximating function we capture the form and coefficients of a polynomial that better characterize the relationship. Using a multivariate approximant may solve the learning problem. The polynomial is obtained such that the maximum absolute error between the data and the approximant is minimized. A polynomial approximant with the GA process can be obtained using any of the techniques described below.

2.4.1 Low frequency spectrum

A low frequency spectrum data might be useful in representing the image content in a lower frequency range when the visible spectrum does not give relevant results.

2.4.2 Gradient spectrum

A gradient spectrum is useful in determining and storing the detected borders of an image through the differences in the levels of gray.

2.4.3 Maximum entropy spectrum

In the experiment [15], three spectra (low frequency spectrum, gradient spectrum, and maximum entropy) for each image were used for face recognition. The recognition of any one face was realized by comparing their three spectra. The study [15] claimed that low frequency spectrum captures “low level” characteristics of a given image used by humans for recognition, gradient spectrum captures borders of an image through differences in the levels of gray, and maximum entropy spectrum tries to reduce the noise in the signal and derive knowledge from any incomplete information. In the experiment performed by

Angel Fernando et al. in [15], the approximating polynomials were set to 12 terms and a highest degree of 6. The combination of three spectra was compared versus the approximation polynomials that characterize the training set as per Mahalanobis distance and nearest match was taken as the best match. The genetic algorithm used is explained in next section.

2.4.4 Vasconcelos Algorithm

Define

n: The number of individuals

P_c*: The probability of crossover*

P_m*: The probability of mutation*

1. Randomly generate a population of size *n*
2. Evaluate the individuals in the population
3. Sort individuals from best to worst
4. While convergence criteria are not met
   a. For $i = 1$ to $n/2$
   b. Generate a random number *r*
   c. If $r >= P_c$ cross individuals *i* and $n-i+1$ (crossover is annular with a ring size I/2)
   d. Randomly mutate ($i * n * P_m$) bits in the population
   e. Evaluate the new *n* individuals
   f. Sort all 2*n* individuals from best to worst
   g. Eliminate the worst *n* individuals.

Some of the findings from the experiment [15] are stipulated below:

2.4.5 Training database

The ORL database, obtained from AT&T Laboratories, was used for training and analysis purposes in the present study. The pictures are in portable graymap (PGM) format with 92 x 112 pixels and 256 levels of gray.

2.4.6 Pre-Processing

Before analysis and training of the faces of ORL database, the images were pre-processed as follows:

1. The average of the gray level was found

2. The total average value ($\mathbf{X}_T$') and a total standard deviation (σ_T) were found

3. The images were equalized using ($\mathbf{X}_T$') and (σ_T) such that each one remained with tones of gray within similar values

2.4.7 Random Sampling of Pixels

A random sample of the pixels is chosen from the faces in ORL such that any selected pixel is used only once. In all, 1,752 pixels (17 percent of 92 x 112 pixels) were used.

2.4.8 Study results using GA

After studying [15] we can draw following results: The GA method does not rely on the identification of face characteristics or template matching before applying the face recognition process. During training and identification relatively small amount of pixels are used for polynomial approximation that eventually reduces the face recognition time. The GA method is independent of other unconstrained environments such as facial variations due to a beard or eyeglasses, etc, unlike other methods that give erroneous results under similar

conditions. The GA method can also be used in the recognition of objects in other images such as arbitrary objects, time series, medical images, astronomical images, etc.

2.5 Elastic Matching

The elastic algorithm solves the problems associated with the variation of results from the face recognition systems that heavily rely on distance measures [29]. Lades et al [33] is based on aspect-graph matching. In simple terms, the Elastic Matching technique generates a model graph from semi-automated system. This model graph is used to compare with the test image.

To apply this algorithm, we need a new type of representation for the face template. Let $\boldsymbol{S}$ be a 2-D image lattice, then the face template is a vector field,

$$\boldsymbol{c} = \{\boldsymbol{c}_i,\ i\ \varepsilon\ \boldsymbol{S}_1\}$$

where $\boldsymbol{S}_1$ is a lattice embedded in $\boldsymbol{S}$ and $\boldsymbol{c}_i$ is a feature vector at position i. The idea is to represent the most critical information about the face. There are various ways to represent the $\boldsymbol{c}_i$ vector. In [14] the author explores the representation using Gabor features.

Let the image be denoted as $\boldsymbol{I}$, and let template c be constructed from the image, defined over S. Let $\boldsymbol{g}$ be a vector whose components (g^1, g^2, g^m) are 2D Gabor filters with various center frequencies, bandwidths, and orientations, given as,

$$\mathbf{g} = [g^1, g^2, \ldots, g^m]^T$$

Then $\boldsymbol{c}_i$ is given by the magnitude of the Gabor filter outputs sampled at the position $i\ \varepsilon\ S_1$.

$$\boldsymbol{c}_i = | (\mathbf{g} * \boldsymbol{I})_i |$$

where * represents convolution, and the absolute value is taken component by component. The Gabor features $\boldsymbol{c}_i$ provide multiscale edge strengths at position i.

Similarly, we can define the observed face as a face vector field on the original image lattice $\boldsymbol{S}$, $\boldsymbol{x} = \{\boldsymbol{x}_j, j \varepsilon \boldsymbol{S}\}$ where x_j is the same type of feature vector as $\boldsymbol{c}_i$ but defined on the fine-grid lattice $\boldsymbol{S}$.

The distance d($\boldsymbol{c}$, $\boldsymbol{x}$) is defined through a “best match” between c and x in the elastic matching approach. A successful match between $\boldsymbol{c}$ and $\boldsymbol{x}$ can be described uniquely through a mapping between $\boldsymbol{S}_1$ and $\boldsymbol{S}$, denoted by

$$\mathbf{M: S_1 -> S}$$

The best match is one that preserves the features and the local geometry. Preservation can be approximated rather than be rigid. Hence, we call this method elastic bunch graph matching.

The quality of different matches can be evaluated using an energy function [31], E(M). The best match is the one with minimum energy. Since large numbers of matches are possible, a simple modified approach is proposed consisting of two steps: rigid matching, and deformable matching.

2.5.1 Rigid matching

In rigid matching, the template $\boldsymbol{c}$ is moved around in $\boldsymbol{x}$, as with conventional template matching. At each position, $||\boldsymbol{c}-\boldsymbol{x}'||$ is calculated, where $\boldsymbol{x}'$ is the part of $\boldsymbol{x}$ such that matching $\boldsymbol{x}'$ with $\boldsymbol{c}$ does not lead to deformation of $\boldsymbol{S}_1$. The match

with the minimum distance position is taken as an initial match M_0 with corresponding energy $\boldsymbol{E}(M_0)$.

2.5.2 Deformable matching

In deformable matching, lattice $\boldsymbol{S}_1$ is stretched through random local perturbations to M_0 to reduce further the energy function. A random local perturbation to M_0 amounts to the following:

a. Pick a point in $\mathbf{S}_1$ at random represented as i.

b. If $M_0(i) = j$, select a random point $j' \varepsilon \mathbf{S}$ that is spatially close to j. Let $M_0'(i) = j'$, where M_0' is the match after a perturbation. For any $i <> \boldsymbol{I}' \varepsilon \mathbf{S}_1$, $M_0'(i') = M(i')$.

c. The distance between $\boldsymbol{x}$ and $\boldsymbol{c}$ is computed when the deformable matching converges to a mapping M* as $d(\boldsymbol{c},\boldsymbol{x}) = E(M^*)$.

2.5.3 Elastic Bunch Graph Matching

To represent an unknown face as a graph we use a labeled graph, the nodes of which refer to points on the object's aspect and are labeled by jets. Edges of the graph are labeled with distance vectors between nodes. Rather than a simple labeled graph that is represented by a jet as a node, we use a data structure called a bunch graph that goes one step further. The elastic bunch graph matching method derives a bunch of jets for each training image and uses the derived bunch of jets to represent the graph node. To form a bunch graph, a collection of facial images is marked with node locations at defined positions of the head. These node locations are called landmarks and are obtained by a semi-automatic process. When matching a bunch graph to an image, the jet extracted

from the image is compared to all jets in the corresponding bunch attached to the bunch graph and the best matching one is selected. This process is called elastic bunch graph matching (EBGM) [29].

A bunch graph covers a variety of faces with different local properties to generate the model graph. The nodes of the model graph are tentatively placed over the test image and jets are extracted from those image points. The extracted jets are used for comparing the model graph with the test image. Varying the node positions in the image optimizes this similarity further. This variation takes the form of a global move of a rigid copy of the model graph's node positions. Later, the image nodes are allowed to move individually, introducing elastic graph distortions. Lastly, using a similarity function, we achieve a significant match between the test image and the model image.

The EBGM algorithm is accomplished in three steps: face finding, landmark finding, and recognition by comparison.

a. *Face Finding* finds a face in an image and determines its size by finding a set of matches to bunch graphs of appropriate pose and of three different sizes. The best matching bunch graph determines the size and position of the face. The identified face image square is warped to a standard size (128 x 128 pixels), and a new wavelet transform is computed, thus defining the image frame. The image frame so identified is given as input to the next phase.
b. *Landmark Finding* repeats the basic step of identifying the important landmark locations with a bunch graph containing more nodes and a larger bunch graph gallery. The landmark finding step also finds facial landmarks

with high positional accuracy and reliability, and encodes the information contained in the image as accurately as possible.

c. *Recognition by Comparison* does pair-wise comparison of the model graphs produced to the test image and obtains a similarity value. The similarity value is computed as the sum of jet similarities between pairs of corresponding nodes divided by the number of pairs, using a phase-insensitive similarity function.

The Colorado State University (CSU) Face Identification system also applies this algorithm. The basic steps followed in EBGM algorithm for the FERET database are: preprocess the FERET images to generate PGM and SFI format images, locate the landmark locations in the rest of the FERET imagery, create a face graph for every image in the database, and lastly, generate distance measures between the face graph and the given probe images.

The elastic bunch graph matching approach is independent of variation in lighting, pose, and expression. In addition to the invariance to internal and external factors, elastic bunch graph matching can assimilate new face template data without any modifications to the existing templates. Gabor features are insensitive to lighting variation, and only features at key points in the image are used, rather than the entire image. The only major drawback of the elastic bunch graph matching technique is that the process is computationally intensive. With the easy availability of more processing power, this drawback may soon be irrelevant.

2.6 Wavelet technique

The wavelet is similar to a sinusoid. It has oscillating wavelike characteristics, but it differs from the sinusoid in that it is a waveform of effectively limited duration that has an average value of zero. This property helps in representing the wavelet more accurately with the local description and separation of signal characteristics. Wavelet is an appropriate mathematical tool to extract the local features of variable sizes, variable frequencies, and at variable locations in an image. Also, the number of wavelet coefficient to represent an image is significantly low. Hence, the wavelet transform is very effective in image compression.

R. Alferez and Yuan-Fang Wang used a modified technique that uses wavelets obtained from color and shape data to recognize the appropriate match [8]. The method used in their study first obtained the geometric characteristics of the object in consideration by processing the object contours, parameterized them, and then subsequently used signature invariant measures to compare the two different objects. By aggregating different derived object characteristics, the system can capture critical information about the object's appearance. Using wavelets and spline bases, the system obtains important coefficients that represent either the object's interior region or the exterior contours.

By normalizing the data, the difference caused due to changes in unconstrained factors such as angle of view, lighting, etc., is eliminated. The steps are given below that obtain the invariant information from the images.

2.6.1 Extraction of Contours

Using a segmentation technique, the contours of the object can be obtained. A segmentation process is a computationally intensive process. To use the system in real time the image database can be analyzed offline to obtain the segmentation information beforehand. However, for the query images, the process can be complicated. To ease some of the constraints, the system can be semi-automated to get input from the user to specify the object of interest.

2.6.2 Parameterization of Contours

A common frame of reference needs to be established before comparing two different image contours. The common frame is comprised of a common starting point and a direction of traversal. A parameterization scheme then traverses corresponding points in the two contours at the same parameter setting.

There are three different techniques that can be used to parameterize the contours of a parameterized curve $c(t) = [x(t),y(t)]^T$. The techniques are:

a. Intrinsic arc length is a very simplistic measure to represent the parameters of contours, but it is not invariant to transforms where shape deformation is allowed.
b. Affine arc length, $\tau = {}_a\int^b$ cube root(x'y''-x''y')dt where *x*', *y*' are the first derivatives w.r.t t and x'', y'' are the second derivatives w.r.t t and (a,b) is the path along a segment of the curve.
c. Enclosed area parameter is the area between two line segments from the centroid of an object to two points, a and b, on the contour.

$$\sigma = \tfrac{1}{2}\ {}_a\int^b |xy'-yx'|dt$$

The affine arc length and enclosed area parameter can be made invariant by normalizing the measure with respect to the total affine arc length or the total enclosed area parameter respectively.

2.6.3 Comparison Matching

If the starting point for the invariant signature differs, then the signatures are a phase-shifted version of each other. If the same contour is parameterized in opposite directions, the invariant signatures are flipped and inverted images of each other. A match can then be asserted as a function of the cross-correlation between the two signatures. The invariant signatures of two contours are used to ensure that the two contours in consideration follow identical traversal directions and have a common starting point. With the use of invariant signatures no point correspondence is required in computing the invariants.

The invariant framework considered in this paper [8] is a general framework that considers variation in an object's image induced by a rigid-body motion, affine deformation, and changes in the parameterization, scene illumination, and viewpoint.

An invariant measure is used to compare two different contour maps. The measure attains the maximum of 1, when the two objects are identical but differ in position, orientation, scale, traversal direction, and starting point.

Interesting results were obtained in [8] that can be used further for developing a reliable face recognition system and are given below.

1. Feature invariant to affine deformation and perspective projection were used to match the silhouette of the query with the silhouette of those in the

database. The illumination invariant was computed in the interior to classify the objects within the model further.

2. For a small set of complex images of objects, 100 percent accuracy of result was shown using a wavelet matching system.
3. To further ascertain the result obtained from the system, the interior regions of the objects were compared for similarity. A characteristic curve were represented for each image and used as a parameter to determine the similarity between the interiors of the two images in consideration.
4. The system was superior in terms of the invariance to changes in the object image induced by the unconstrained environment factors and was powerful enough for within-class retrieval.

2.7 Importance of Color Information

Preprocessing images to make them invariant of color and other illumination information is a common step in almost all the face recognition techniques. R. Alferez and Yuan-Fang Wang [8], and L. Torres et al. [18] have explored the importance of color information in the face recognition process. A modified technique using shape and color information as deciding criteria is proposed [8] to index images in databases. We will discuss the importance of color information and other spectral representations of the image.

There are some common techniques used to represent color information (e.g. RGB, YUV, and HSV models [8]). We will list the result of the various color modeling techniques that we studied.

1. The recognition rate achieved by using only luminance Y (from YUV color model), RGB model or independent components from RGB model gave the same results.
2. Using Y, U, and V components simultaneously instead of an independent Y component gave roughly a 4-5 percent improvement in results. Similar results were obtained for S and V component combinations from the HSV color model.
3. A color space with a separation of the luminance and chrominance information tends to provide better results than a color space with this information mixed (as in RGB). Some color components carry more information than was useful in face recognition.
4. In YUV color, maximum information was held in the Y component.
5. In general, by adding the color information, the recognition rate increased.

Another aspect of face recognition has been explored using visible and thermal infrared imagery. Diego A. Socolinsky and Andrea Selinger used such a technique [21] and the findings are given below.

The (LWIR) Longwave Infrared Imagery is explored, in the spectral range of 8μ – 12μ. To reduce the inconsistent illumination factor that gives erroneous results in face recognition, pre-processing steps are applied. The methods available are histogram equalization, Laplacian transforms, Gabor transforms, logarithmic transforms, and 3D shape based methods. Due to the large variation in illumination, the within-class variability introduced is shown to be

significantly larger than the between-class variability in the data, thus severely affecting the classification performance. Thermal infrared imagery of faces, however, is almost invariant to changes in ambient illumination.

The data used to obtain the results was acquired with a newly developed sensor capable of capturing simultaneous co-registered video sequences with a visible CCD array and LWIR micro-bolometer.

Appearance based methods have generally shown higher performance than those based on facial geometry alone. Projection of the image into a subspace of the image space followed by 1-nearest neighbor classification was used in all the algorithms. The algorithms tested using the color information in the paper [2], [18] and [21] were:

2.7.1 EigenFaces (PCA)

The face space is computed by taking a set of training observations. This training set is then used to find the unique ordered orthonormal basis of the data space that diagonalizes the covariance matrix of those observations, ordered by the variances along the corresponding single dimension subspace. These vectors are known as the principal components, or Eigenfaces. For a fixed choice of *n* basis vectors, the subspace spanned by the first *n* basis vectors is the one with the lowest reconstruction error L^2 for any vector in the training set used to create the face space. The face space is taken to be a representational model for all the sets of face images in low dimension.

2.7.2 Linear Discriminant Analysis (LDA)

Under the assumption using Gaussian distributed classes having the same variance and linear separability, one can show that the optimal subspace in which to perform classification is spanned by the solution vectors w of the following generalized eigenvalue problem $\mathbf{S}_b w = \lambda \mathbf{S}_w \mathbf{W}$, where $s_{w,}$ s_b are the within-class and between-class scatter matrices respectively.

2.7.3 Local Feature Analysis (LFA)

Another method used to explore face recognition is based on minimizing the correlation between the basis vectors obtained by enforcing topographic indexing of the basis vectors for the facial data based on the second order statistics results. Local Feature Analysis achieves this by constructing a family of feature detectors based on PCA decomposition that are locally correlated. A selection or sparsification step is then used to produce a minimally correlated subset of features that define the subspace of interest.

2.7.4 Independent Component Analysis (ICA)

With the help of a non-orthogonal basis, ICA tries to achieve the marginal data that is statistically independent. Computing independent components is not accomplished by solving an algebraic system of equations, but by numerically minimizing a criterion function.

Considering the effect and results of LWIR on ICA based face recognition technique [21], we summarize following points that might be useful in our thesis work:

1. Results on visible imagery are always inferior to those on LWIR imagery.

2. Recognition performance on visible imagery, regardless of algorithm is worse for pairs where both illumination and facial expression vary between the training and testing sets, followed by pairs where either illumination or expression differ.
3. The worst performance for LWIR recognition occurs for the images having a change in facial expression implying a change in shading as a result of the varying surface normal.
4. The LWIR imagery of human faces is not only a valid biometric, but also significantly superior in comparison with the visible imagery.
5. Preliminary results on fusion of modalities are extremely promising, indicating that a further reduction of error of 50 percent over LWIR performance may be possible.

2.8 Miscellaneous/Hybrid techniques

The most promising techniques from the above discussion are Eigenface, Elastic Bunch Graph Matching, Gabor wavelet matching, and Genetic Algorithm techniques. A few other interesting techniques have been proposed that can be used for face recognition in [2], [8], [9], [15], [10], [16], [14], and [19].

In this section we will discuss a few of these techniques.

2.8.1 Mixture of Principal Components (MPC)

Modified techniques have been proposed that use PCA in combination with other features such as [18] and [14]. Deepak Turuga and Tsuhan Chen used mixtures of principal components as a technique to model facial information.

They proposed an efficient statistical modeling technique based on linear extension to the traditional Principal Component Analysis (PCA) called Mixture of Principal Components (MPC) [9].

As stated before, the template matching technique gives inconsistent results due to the lack of freedom in terms of orientation and complexity of unconstrained environments such as shape, texture, pose, illumination variations, etc. The simple PCA technique does not give accurate results especially when the data consists of multiple clusters or extreme amounts of variation exist in the data. Hence, a modified PCA version is proposed, PCA with extension.

2.8.1.1 Non-linear approach

The non-linear approaches are Multi-Dimensional Scaling (MDS), Locally Linear Embedding (LLE), and principal surfaces as an alternative to PCA proposed by Hastie and Stuetzle [32].

All the non-linear techniques are computationally intensive and also lack an easy forward-backward transformation.

2.8.1.2 Linear approach

a. **Vector Quantization (VQPCA)** [33], data samples are partitioned into clusters based on which cluster reconstructs them with the smallest error. The parameters of each cluster are then updated using local PCAs, and this process is iterated till the convergence of parameters. This leads to the loss of global information present in the data.

b. **Probabilistic PCA (PPCA)** uses a mixture of PPCA to represent the data. The squared error is not minimized. Instead, the likelihood of observing the data given the model is maximized during dimensionality reduction. Construction performance of the model is poor.

c. **Mixture of Principal Components (MPC)** automatically models the data using a mixture of eigenspaces. The MPC parameters are chosen to minimize the overall reconstruction error instead of optimizing the likelihood of observing the data for the given model.

The MPC model is characterized by two sets of parameters, the means, and the eigenvectors of the component eigenspaces. The basic approach to reconstruction is the linear combination of individual reconstructions from a mixture of component eigenspaces.

Given a data set vector $\boldsymbol{y_{ij}}$, project it onto each of the component eigenspaces to obtain individual reconstructions $\boldsymbol{y'_{ij}}$. Linearly combine these individual reconstructions to obtain the representation that is closest to the original data vector $\boldsymbol{y_j}$. The individual reconstruction for test vector *i* from mixture component *j* is obtained as, $y'_{ij} = m_j + \Sigma_k^p[(y_i - m_j)^T u_{jk}] u_{jk}$. The individual reconstruction is linearly combined using a set of weights. The weights are solved individually for each of the test vectors with a constraint that the summation of all weights must be one. After reconstruction, we use an iterative expectation maximization algorithm to train and determine the mixture means and eigenvectors given a set of training data. The training problem is formulated as a minimum error optimization problem. Overall, the MPC technique showed

smaller representation error than the PCA for data with large variations in appearance. MPC had around 34 percent smaller reconstruction error for faces with varying poses and under different illumination conditions, even with the same number of total eigenvectors for both the models, and can be very useful in capturing data variations in any generic data set. MPC is a general statistical modeling tool, and had a recognition performance of 95.8 percent as compared to the PCA with only 83.8 percent. Better results can be obtained by progressive training and by use of a model with data compression capabilities.

2.8.2 AI Learning Model with Gabor/DoG filtering

The majority of facial recognition systems solve the recognition problem by reducing the data to a lower dimension. By removing unnecessary information, face recognition efficiency is increased significantly without loss of accuracy. There are various techniques that concentrate on this approach, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), Gabor filters and various iso-density map or feature extraction schemes.

In the AI Learning Model with Gabor/DoG filtering technique, Radial Basis Function (RBF) is explored [10]. The main characteristics of this function are computational simplicity and statistical robustness.

2.8.2.1 RBF Network Model

The RBF network model is a two-layer hybrid-learning network consisting of an unsupervised learning layer and a supervised learning layer. The unsupervised learning layer handles processing from the input to the hidden units, where individual radial Gaussian functions from each hidden unit simulate

the effect of the overlapping and locally tuned receptive fields. The supervised layer continues processing from the hidden to the output units. Each function has an associated center width value that defines the nature and scope of the unit's receptive field response, giving an activation that is based on the relative proximity of the test data to the training data.

The weights can be adjusted using the Widrow-Hoff delta learning rule or the single layer of linear output units that allows a pseudo-inverse matrix method for their exact calculation. The pseudo-inverse matrix method completes the learning phase in a fraction of a second.

Two image sequences were used to simulate an unconstrained environment and primary and secondary image sequences. Primary image sequences are useful for training the system to build on-line probing. These consist of image sequences of a person with constrained properties such as a controlled background and a pre-determined location of the person in the image. Secondary image sequences capture unconstrained environment such as the uncontrolled movement of a person with differing backgrounds.

2.8.2.2 Pre-Processing of Segmented Data:

Two main pre-processing techniques are considered before applying RBF: 1. The Difference of Gaussian (DoG) filter, and 2. Gabor wavelet analysis.

The pre-processing of segmented data technique is analogous to the visual neurons whose receptive field is the area of the visual field. For example, a DoG filter can be used to implement the receptive fields. The receptive fields so developed are similar to the one developed in the retinal ganglion cells and

lateral geniculate cells in the early visual processing stage. By ensuring temporal coherence, high confidence in a ‘time window’ can be utilized in periods of low-confidence output.

2.8.2.3 Study results using RBF network model

1. The RBF network was shown to generalize well from the samples in the classifying faces from real time sequences. Gabor preprocessing was shown to give a more useful input representation than the DoG preprocessing, especially for the more difficult secondary sequence. For a small training set, handling sequences was more important in discarding uncertain classifications with the use of the confidence measure. The locally tuned linear RBF networks showed excellent performances in simpler face recognition tasks when trained and tested on images from primary sequences.

2.8.3 Image warping with FTSM.

A hybrid method for detecting varying face poses is explored along with the facial feature registration using color images.

Face feature registration is a two-step process: 1. Identify the face in the given image using a skin color Gaussian model, and 2. Compare each face candidate with a varying pose face model using a combined feature-texture similarity measure (FTSM).

The face recognition techniques that use facial characteristics for face recognition can give misleading results due to the unconstrained values for various factors (e.g. face appearance, lighting conditions, and facial

expression). In fact, even a seemingly simple task like identifying a face in the given image, is difficult due to the above stated reasons.

To reliably detect human faces, the system must either be insensitive to the variations encountered, or it has to take these factors into consideration. The PCA method relies heavily on the distance measure and analysis of the variation in the training images provided to the system, rather than the facial information in general. Thus, PCA and similar techniques will give inaccurate results for images with significantly high variation in expression, lighting, and/or pose.

To make the system more robust and invariant to a large change in the above factors, a non-linear model that combines texture and pose differences using the image warping technique has been proposed by Lixin Fan and Kah-Kay Sung in [2]. The system can be trained to learn from the training faces and then use this knowledge to formulate face detection, a type of model-based image matching problem. Another, simpler, method uses a statistical color model, which captures variations of different skin colors for face detection but this method can give lots of false alarms. Also, determination of facial features through color information is difficult to achieve.

Face registration can be narrowed down easily by rejecting the areas that do not contain any color information corresponding to a skin region. Brightness normalized color space and a Gaussian color distribution model are used to locate skin color regions in an image by first normalizing color components by dividing the red and green components by the intensity. A pre-trained Gaussian classifier covering various spectrums of skin color is then used to classify the

pixels that might potentially contain facial information. Identified regions are further narrowed down with the help of a threshold area that can possibly represent the face region. Facial regions are then compared to perform face registration.

A combined Feature Texture Similarity Measure (FTSM) combines linear subspaces of face texture and structural difference [2]. A combined FTSM gives quantitative difference between a given image and a model image. To represent a varying pose face model, the method first decouples textural and structural image variation and then combines them using a compound varying pose face model.

2.8.3.1 Decoupling Process

2.8.3.1.1 Preprocessing

*A **face prototype*** is defined as the one that consists of 16 line segments and 24 end points of line segments. This prototype is the mean shape of all the example face shapes obtained by a manual registration of example face images. The line segments and end points represent facial features including eyes, nose, and mouth.

Feature-based image warping is used to warp the example face images with respect to the face prototype. Warped example face images have the same structure as the face prototype.

The preprocessing step generates two types of example data:

a. Structurally normalized face images that have textural variation only and do not have the structural variations information.
b. A set of feature point correspondence maps (FPCM) that represent the structural variation between the training face image and the prototype face. Mathematically, an FPCM can be represented as a $2\boldsymbol{N_e}$ dimensional column vector.

$$\mathbf{C} = [dx_1\ dy_1\ \ldots dx_{\boldsymbol{Ne}}\ dy_{\boldsymbol{Ne}}]^{\mathrm{T}}$$

where $(dx, dy) = (x_e, y_e) - (x_p, y_p)$ represent the disparity of corresponding feature points (x_e, y_e), (x_p, y_p) in two images.

1. ***Textural variation modeling*** applies PCA technique to obtain eigenfaces from the structurally normalized face images. The eigenfaces so obtained can be used to represent the textural information variation in the face images. Mathematically, $\boldsymbol{I}$ = (approx) = $\boldsymbol{I}'$ = $\mathbf{X}\ \boldsymbol{r}$, where $\mathbf{X}$ is an eigenvector matrix of retained significant eigenvectors and $\boldsymbol{r}$ is the transformation vector determined by $\boldsymbol{r} = (\mathbf{X})^{\mathrm{T}}\boldsymbol{I}$, Elements of $\boldsymbol{r}$ are referred to as texture parameters. Eigenvectors with maximum information are retained and the eigenvectors whose sum of the eigenvalues is significantly lower than the maximum eigenvalue are eliminated. A detailed analysis confirms that the more significant eigenvectors approximately represent the change in lighting conditions, and the less significant eigenvectors capture the face appearance differences between individuals.
2. ***Structural variation modeling*** applies PCA on example FPCMs to get a set of eigenvectors, referred to as the eigenflow in the sequel, and uses the

eigenfaces to represent the structural variation in the face images. Mathematically, $\boldsymbol{C}$ = (approx) $\boldsymbol{C}'$ = $\mathbf{Y}\boldsymbol{s}$, where $\mathbf{Y}$ is an eigenvector matrix of retained significant eigenvectors and $\boldsymbol{s}$ is the transformation vector determined as $\boldsymbol{s} = (\mathbf{Y})^T\boldsymbol{C}$. Elements of γ are referred as pose parameters. Given an initial image I_0 and a FPCM $\boldsymbol{C}$, a new image can be synthesized by using the feature based image warping technique as, $\boldsymbol{I} = I_0 \mathbf{o} C$ where **o** denotes the warping process that translates the image pixels according to given FPCMs $\boldsymbol{C}$.

3. ***Combining textural and structural models:*** An unknown face image can be represented using the structural and textural model as, $\boldsymbol{I}$ = (approx) $\boldsymbol{I}^M(r, s)$ = ($\mathbf{X}$ r) **o** ($\mathbf{Y}$ s). The image warping technique causes a non-linear transformation of frontal face images and results in a complex manifold in the image space.

2.8.3.2 Combined FTSM

Using a similarity measure we can quantify the similarity between the model approximations I^M, and the varying pose face identified with the image pattern I at candidate locations. To represent the similarity measure, a combined FTSM is used that combines the pose aligned intensity difference component reflecting textural differences and a feature based similarity measure component that represents structural differences between the two images $\boldsymbol{I}$ and $\boldsymbol{I}^M$.

$$Sim(\boldsymbol{I},\boldsymbol{I}^M) = (S_{feature} + S_{texture})$$

where $S_{texture} = (1/N) \Sigma (I_i - I^M_i)^2$, i = 1..N and

$S_{feature} = (1/N_e) \Sigma (d_i)^2$, i =1..N_e and

N is the number of image pixels and

$d_i = dx_i^2 + dy_i^2$ is the length of the displacement vectors and N_e is the number of feature points.

2.8.3.3 Pose Face registration

Face registration can be achieved by using a model based image-matching problem. Optimal texture and pose parameters are found when the FTSM between the original image I, and the model image I^M is minimized.

A numerical solution is easier to obtain rather than an analytic solution. First the registration algorithm starts with the frontal face image prototype as a basemark. Align two given images, and synthesize textural variation. This variation is caused due to changes in the object's appearance and lighting conditions. By re-estimating textural variations repetitively more reliable extraction of feature points are obtained. After the final match is found we compute the optimal FPCM based on the estimated pose parameter g* using [3] and apply it to the prototype feature points to obtain the warped feature points that are aligned with the given face poses.

2.8.3.4 Face verification

The minimized FTSM is compared with a preset threshold to decide the presence of a face pattern in the given image. By analysis of skin color patterns for faces, we obtain the appropriate value for the threshold.

2.8.3.5 Results from Image warping technique study

The statistical techniques give erroneous results for the face images that have extensive variation in terms of expression, pose, lighting, etc. While, face pose detection can be reliably obtained using correspondence maps between the varying pose images. A naïve face detection method based on color based detection in combination with elliptic masks application is subject to lot of false alarms, but with the usage of FTSM threshold, false alarms can be totally eliminated. Using both textural and structural information in a combined pose face model, gives better results for the face pose detection and registration than a simple color detection technique. Optimal pose parameters can be obtained by learning a mapping between pose parameters and face pan tilt angles. This information can be used successfully to obtain pose information in the given image.

2.8.4 Three dimensional Morphable Techniques

The 3D morphable model techniques try to model a 3D morphable face that encodes shape and textures in terms of the model parameters. The algorithm that recovers these parameters from a single image of a face is also discussed later in this section. Most image analysis based techniques try to fit the generative model to a novel image, thereby obtaining appropriate model parameters. To make identification independent of imaging conditions, the goal is to separate intrinsic model parameters of the face from extrinsic imaging parameters [23]. Most of the techniques that use such an approach are statistically oriented techniques. Another closely related technique uses synthetic models that are

generated from the training image, and this synthetic model is later used as a base to determine the best match. In the 3D morphable technique, a 3D computer graphics technology is used to form a 3D image. Using an algorithm that tries to fit the model to the given image the process finds the equivalent parameters. A similar technique is used in [19]. A 3D color camera is used to obtain a 3D image, and hence, the computation required to generate a 3D model is eliminated at the cost of the equipment used to capture a 3D image.

In this approach the estimate is achieved by fitting a statistical, morphable model of 3D faces to images. The model is learned from a set of textured 3D scans of heads.

A 3D laser scan is used to obtain the scan of an object that is represented as,

$$\mathbf{I}(h,\phi) = (r(h,\phi),\boldsymbol{R}(h,\phi),\boldsymbol{G}(h,\phi),\boldsymbol{B}(h,\phi))^{T},$$

where h = vertical steps, ϕ = angular steps, and r = radius.

The core step of building a morphable face model is to establish a dense point-to-point correspondence between each face and a reference face, which can be a scan from the database or any other 3D face model.

The vector field gives dense correspondence,

$$\mathbf{V}(h,\phi) = (\triangle h\ (h,\ \phi),\ \triangle\phi(h,\ \phi))^{T}$$

such that each point $I_1(h,\phi)$ in the first scan corresponds to the point $I_2(h+\triangle h,\ \phi+\triangle\phi)$ in the second scan.

To find this vector field, the optic flow algorithm was extended from the grey-level images $\boldsymbol{I}(x,y)$ to the vector valued arrays $\mathbf{I}(h,\ \phi)$, replacing the products of grey values $I_1(x,y)\ .\ I_2(x,y)$ in the algorithm by scalar products $<I_1,$

$I_2> = wrr_1r_2 + w_RR_1R_2 + w_GG_1G_2 + W_BG_1G_2$ with weight factors wr, w_R, w_G, W_B that compensate for different variations within the radius and texture data. The coordinates and texture values of all n vertices of the reference face are concatenated to shape and texture vectors.

$$S_0 = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T$$

$$T_0 = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T$$

Vector S_i and T_i of the examples i=1..m in the database is formed in a consistent way using the flow field $v(h, \phi)$ from the reference face to face.

Convex combinations of the examples produce the novel shape and texture vectors S and T.

We can represent

$$S = \Sigma_i^m \; a_i S_i \;, \; i = 1..m$$

$$T = \Sigma_i^m \; b_i T_i \;, \; i = 1..m$$

Such that a_i and b_i should sum to 1 so that the changes in overall size and brightness are avoided.

The system performs PCA separately on the shape and texture vectors S_i and T_i, and ignores the correlation between shape and texture data.

For shape, subtracting the average s',

$a_i = S_i - s'$, with $s' = 1/m \; \Sigma_i^m \; S_i$,

A = data matrix = $(a_1, a_2, \ldots, a_m)$ then the covariance matrix **C** is obtained as **C** = 1/m $\mathbf{AA}^T$ by a singular value decomposition of A.

The eigenvectors form an orthogonal basis,

$$S = s' + \Sigma_i^{m-1} \; \alpha_i \; .s_i,$$

$$T = t' + \Sigma_i^{m-1} \beta_i . t_i$$

In an analysis-by-synthesis loop, the morphable face model can be fitted to a novel face shown in an input image $I_{input}(x,y)$. The goal of the image analysis is to find the model parameters α_i and β_i, and the face position, orientation and illumination such that the model, rendered by the computer graphics algorithms, produces an image as close as possible to the input image. Three steps are taken t o achieve this: 1. Image synthesis, 2. fitting the model to an Image, and 3. Identification criterion.

2.8.4.1 Results from 3D Morphable technique study

In spite of the large variations in illumination and the difference in the viewpoint from the front to the profile, the performance of the algorithm used was significantly positive. The model parameters of shape and texture are an appropriate representation of the identity of a face. Higher identification performance on the entire set can be achieved by increasing the reliability of the fitting algorithm.

2.9 Conclusion

In this review of the current literature we explored various face recognition methods. Among them, the most promising were statistical techniques (PCA and ICA [1], [3], [4], [5], [6] and [7]), wavelet matching (Gabor wavelet techniques [8] and [16]), and an Elastic graph matching technique in [14], an EBGM technique in [29] and Genetic Algorithm [16] and [15]. There are other modified techniques such as Mixtures of Principal Components [9] that are worth

exploring further. The choice in using PCA or ICA technique is dependent on the problem at hand.

For a face recognition technique, ICA architecture II with FastICA is more efficient than PCA (used with the L1 or the Mahalanobis distance measure). However, to find face expression efficiently, ICA architecture I with InfoMax generates best results.

In contrast to the traditional technique, there are techniques that give positive results. Some of the techniques are the one based on the Feature-Texture Similarity Measure (FTSM) that combines the texture and shape data [2], and a similar technique using Gabor wavelets for representing shape and color data [8]. Also, the technique using Genetic algorithm in combination with the Gabor wavelet technique [16] and other techniques using multivariate analysis with GA in [15] presented interesting approaches to solving the face recognition problem.

The performance of auto association and classification nets is upper bounded by that of the eigenface approach and is more difficult to implement in practice as in [14]. We now summarize the Eigenface and the Elastic Bunch Graph Matching techniques.

2.9.1 Eigenface

Advantages:

1. Eigenface is computationally less intensive.
2. It gives a good performance with significantly “low” variation in the lighting, face position, and expression.

Disadvantages:

1. The Eigenface algorithm performance deteriorates when the lighting variation, face position, and expression change significantly.
2. The Eigenface approach needs a preprocessing step that scales, positions and compensates for the possibility of extreme variation in lighting, face position and expression.
3. The Eigenface approach may need to modify the existing database information when new data is added to the database.

2.9.2 Elastic Bunch Graph Matching

Advantages:

1. The elastic bunch graph matching is independent of the variation in lighting, face position, and expression.
2. The EBGM algorithm easily assimilates new face template data without modifications to the existing templates. Gabor features are insensitive to lighting variation and only features at key points in the image are used rather than the entire image.

Disadvantages:

1. In comparison with other commonly used techniques, elastic matching is more computationally intensive.

2.9.3 Proposed Comparison test

Having considered the theoretical comparison of the available face recognition techniques, we can infer rough ranking among the techniques and advantages/disadvantages of the potential techniques. We concentrate our study

of face recognition on two specific techniques, PCA technique (Algorithm 1) and EBGM technique (Algorithm 2). Genetic Algorithm is a promising technique too, but is beyond the scope of this research. Non-availability of multiple data sources to compare PCA and EBGM technique further encouraged the present comparison study.

To conduct our experiments we will be using FERET images (explained in section 3.2) and Face Identification and Evaluation (FaceIdEval) system provided by CSU on Redhat Linux platform. We will also give brief description of the features provided by FaceIdEval system later in section 4.1. In this thesis paper experiments, we first studied different parameters that affect the PCA and the EBGM technique, and later conducted experiments to compare the PCA and the EBGM technique. In experiment I, we compared the affect on PCA recognition rate with the choice of different distance measures. In experiment II, we compared the recognition rate as a function of different cutoff methods available with the PCA technique. Choice of eigenvector size and its affect on recognition rate on PCA technique was studied in experiment III. In experiment IV we concentrated our study on the EBGM technique, specifically studying the affect on recognition rate with the choice of wavelet and distance measure. In experiment V, we compared the EBGM and the PCA technique over various probe and gallery set.

3 EXPERIMENTAL SETUP

3.1 System Specifications

The system specification used for the experimental setup is given below:

Redhat Linux 9.0 (Kernel 2.4-21.4)

256 MB Ram

Pentium® 4 CPU 1.70 GHz

30GB Hard Disk

CSU Face Identification Evaluation System (version 5.0)

FERET Facial Image Database Release 2, March 2001

3.2 FERET Database

The FERET data contains 14,051 eight-bit grayscale images provided on a two CD set. The FERET data captures human head images ranging from frontal to left and right profiles of which only 3,816 represent frontal images [28]. The FERET data is available for research purposes from NIST [38]. The images are mainly categorized as gallery, and probe sets. Face recognition algorithms search the given probe image in the gallery, and retrieve rank zero image as the best matching image. More than one algorithm can be compared using cumulative match scores using the ranking results obtained for various probe sets. Various common gallery and probe sets that were used by several face recognition methods are shown below in Table 1. This gallery and probe set provided by the FERET is taken from [38].

Partition	Count	Description
Training set	501	Training images are roughly 80% from FA partition and 20% from DUPLICATE I images
FA gallery	1196	Images taken with one of two facial expressions: neutral versus other.
FB probe	1105	Images taken with other facial expression.
FC Probe	194	Subjects taken under different illumination.
DUPLICATE I probe	722	Subjects taken later in time.
DUPLICATE II probe	234	Subjects taken later in time, this is a harder subset of dup I.

Table 1: FERET gallery and probe set.

We briefly explain the gallery and probe sets below:

FA is a searchable Gallery set consisting of 1,196 images.

FB probe set contains 1,195 images of different individuals capturing either neutral or other expressions. For every individual two distinct expression images are taken, one of the images is placed in the FB probe set and the other is placed in the gallery set.

FC, In comparison to the gallery set images FC probe set consist of images that are taken under different illumination. FC probe set contains 194 images.

DUPLICATE I probe set consists of 722 images satisfying the constraint that gallery and probe images are different. The DUPLICATE I (Dup I) probe set has images taken under varying time duration, ranging from same day to a year apart.

DUPLICATE II probe set consists of 234 images that are taken at least one year after the corresponding image in the gallery. The gallery consists of 864 images in which the DUPLICATE II (Dup II) probe set image is searched.

Percentage of Images in Common

	Train	Fa	Fb	Dup I	Dup II	Fc
Train	100.00%	33.11%	0.00%	13.71%	0.00%	0.00%
Fa	79.04%	100.00%	0.00%	0.00%	0.00%	0.00%
Fb	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%
Dup I	19.76%	0.00%	0.00%	100.00%	100.00%	0.00%
Dup II	0.00%	0.00%	0.00%	32.41%	100.00%	0.00%
Fc	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%

Table 2: Percentage overlap between gallery and probe images.

For FB, and FC probe set we use the whole FA as the gallery set, while to search a DUPLICATE I, or DUPLICATE II probe set we only use 864 images of the FA gallery set.

Percentage of People in Common

	Train	Fa	Fb	Dup I	Dup II	Fc
Train	100.00%	35.37%	35.40%	61.73%	64.00%	0.00%
Fa	98.83%	100.00%	100.00%	100.00%	100.00%	100.00%
Fb	98.83%	99.92%	100.00%	100.00%	100.00%	100.00%
Dup I	35.05%	20.32%	20.33%	100.00%	100.00%	0.00%
Dup II	11.21%	6.27%	6.28%	30.86%	100.00%	0.00%
Fc	0.00%	16.22%	16.23%	0.00%	0.00%	100.00%

Table 3: Percentage overlap in people common between gallery and probe set.

Percentage overlap [28] between the gallery and probe sets in terms of images and individuals is shown in Table 2 and Table 3, respectively.

4 EXPERIMENTS AND RESULTS

The FERET system developed at NIST was initially used to obtain the experimental results. See table 1 for a description of the FERET database categories. PCA algorithm is provided as part of the FERET system, but the EBGM technique is not available in the FERET. The EBGM algorithm with source code is provided by Colorado State University Face Identification Evaluation system (CSU FaceIdEval). We had lot of technical difficulties in compiling and using the FERET system under our development platform. In addition, the algorithms provided on FERET Release 2 are not well updated. After conversing with the technical person [37] in FERET to find possible solution to the compiling and other problems we were facing, we were advised to conduct all the experiments using CSU FaceIdEval [28]. Due to the latest algorithms and tools that are offered with the CSU system, this tool was found to be more appropriate for our experiments. The CSU system provides both the EBGM technique and the PCA technique.

4.1 CSU Face Identification Evaluation System:

The CSU Face Identification Evaluation system is released with an open source license format. The CSU system was easy to install on the experimental setup platform, and accepts image files that are presented in Portable Graymap (PGM) format. We wrote a script to convert the FERET images from compressed Tagged Image File Format (TIFF) to PGM format (see Appendix 6.1). The processed TIFF files either generated PGM or Portable pixmap (PPM) files. The PPM files were later converted into PGM format files.

4.1.1 Algorithms:

We conducted experiments to verify the PCA algorithm findings that we studied in previous sections and explore the EBGM algorithm. Although, the CSU system provides variety of scripts to conduct EBGM analysis there is not sufficient documentation available. However, PCA, PCA+LDA, and BIC algorithms are provided with sufficient documentation. The CSU system provides four base-line face recognition algorithms:

a. A standard PCA, or Eigenfaces algorithm.
b. A combination PCA and LDA (PCA+LDA) algorithm based upon the University of Maryland algorithm in the FERET tests.
c. A Bayesian Intrapersonal/Extrapersonal Image difference Classifier (BIC) based upon the MIT algorithm in the FERET tests.
d. An Elastic Bunch Graph Matching Algorithm (EBGM) that uses localized landmark features represented by Gabor jets. This algorithm is based upon the USC algorithm in the FERET tests.

4.1.2 Analysis Tools:

The CSU system provides two basic analysis tools, the first generates cumulative match curves of the type commonly reported in the FERET evaluation and the later Vendor Test 2000 and is called ***csuAnalyzeRankCurve***. The second tool performs a permutation of probe and gallery images analysis to generate a sample probability density curve for recognition rates at different recognition ranks and is called ***csuAnalyzePermute***.

The csuAnalyzeRankCurve tool generates two text files suffixed with Curve.txt and Images.txt. Both files contain ASCII tabular data that is tab delimited. For, say, feretFAFB file prefix, two files are generated feretFAFB_Images.txt and feretFAFB_Curve.txt. A sample Images.txt suffixed file is shown below in Table 4. The first column contains the file names of the probe images in the analysis. The remaining columns summarize the recognition rank data for each distance measure being studied. Actual distance measures used as fields are: distances/feret/ENERGY_PCA_MahCosine distances/feret/EBGM_Base_PredictiveStep. The values in column 2, 3, etc., of Table 4 are interpreted as follows. Zero means the best match is of the same subject. Likewise, a one means there is one gallery image of another subject that is closer than the gallery image of the same subject, and so on. This value is commonly called the recognition rank. Hence, lower rank means better match for the given probe and gallery image.

ProbeName	ENERGY_PCA_MahCosine	EBGM_Base_Pstep
00001fb010_930831.sfi	0	0
00002fb010_930831.sfi	0	0
00003fb010_930831.sfi	0	0
00004fb010_930831.sfi	0	0
00005fb010_930831.sfi	0	0
00006fb010_930831.sfi	0	0
00007fb010_930831.sfi	0	0
00008fb010_930831.sfi	0	0
00009fb010_930831.sfi	0	0
00010fb010_930831.sfi	0	0
00011fb010_930831.sfi	0	0
00012fb010_930831.sfi	0	0
00013fb010_930831.sfi	0	0
00014fb010_930831.sfi	0	0
00015fb010_930831.sfi	0	0
00016fb010_930831.sfi	0	0
00017fb010_930831.sfi	6	17
00018fb010_930831.sfi	0	0

00019fb010_930831.sfi	0	0
00020fb010_930831.sfi	0	0

Table 4: Top 20 rows of the feretFAFB_Curve.txt are shown above.

For example, in the above Table 4, 00017fb010_930831.sfi image with MahCosine distance measure on the PCA obtains the closest match at position 6, while the EBGM method obtains the closest match at position 17. Lower rank means nearer the best matching image, hence, for this particular probe image, PCA does better than compared to EBGM technique.

The recognition rank file such as feretFAFB_Images.txt also provides some insight into how the cumulative match curves are generated. To compute the recognition count, i.e., how many images are correctly recognized by going a particular depth into the list of sorted gallery images, all that is needed is to scan down a column and count how often the recognition rank is less than or equal to the desired depth. So, to compute the recognition count for the PCA algorithm using the Mahalanobis Cosine distance measure at rank one, scan down the column counting how often a zero appears. For recognition count at rank two, count how often a value less than two appears, and so on. Recognition count is the actual number of probes images correctly recognized at a given rank, while recognition rate is the recognition count divided by the total number of probe images.

The second file generated by csuAnalyzeRankCurve, which for this example is feretFAFB_Curve.txt is shown in Table 5. The Table 5 shows the first 10 rows of this file. The first row contains column headers. The header of the first column indicates this is the recognition rank, starting at zero and running

through the number of probes in the probe set. The remaining columns come in pairs, a recognition count and a recognition rate for a given distance measure at the indicated recognition rank. In Table 5, these column headers have been abbreviated for convenience. We can easily observe different recognition rates for each algorithm at a given rank. For example, in Table 5, PCA with Mahalanobis Cosine distance at Rank 0 generate 73.7 percent recognition rate that is better compared to 73.4 percent produced by EBGM. It is worth noting that EBGM algorithm recognition rate improves as the recognition rank increases.

Rank	PCA MahCos	PCA_MahCos	EBGM_Ptep	EBGM_Pstep
0	472	0.737500	470	0.734375
1	495	0.773438	489	0.764062
2	507	0.792188	505	0.789062
3	521	0.814063	515	0.804688
4	523	0.817187	528	0.825000
5	527	0.823438	537	0.839063
6	528	0.825000	545	0.851562
7	533	0.832812	549	0.857812
8	535	0.835938	551	0.860938
9	538	0.840625	553	0.864062
10	541	0.845313	558	0.871875

Table 5: First 10 rows of the feretFAFB_Curve.txt file.

The CSU permutation of probe and gallery images tool, csuAnalyzePermute, performs virtual experiments using distance files. By taking random permutations of the probe and gallery sets and then performing nearest neighbor classification. It then generates a sample probability distribution for recognition rate under the assumption that probe and gallery images are interchangeable for subjects.

The csuAnalyzePermute tool computes three files for each distance measure along with a summary file. So, for example, the files Energy_PCA_MahCosine_HistCounts.txt, Energy_PCA_MahCosine_HistProbs.txt and Energy_PCA_MahCosine_CMCurve.txt are generated for the distance file produced by the PCA algorithm using the Mahalanobis Cosine distance. The resulting files are tab-delimited tables in ASCII format, and Table 6 and Table 7 show portions of the files for Energy_PCA_MahCosine_HistCounts.txt and Energy_PCA_MahCosine_HistProbs.txt, respectively loaded in Microsoft excel.

The top rows in Table 6 and Table 7 are the same and are column headers. Likewise, the first two columns are identical. The first column, rc, is an exact recognition count running from 0 to the number of probe images. The second column is the recognition rate: recognition count divided by the number of probe images.

1	rc	rr	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10	r11	r12	r13	r14	r15	r16	r17	r18	r19	r20
98	96	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
99	97	0.61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	98	0.61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101	99	0.62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
102	100	0.62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
103	101	0.63	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
104	102	0.64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
105	103	0.64	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
106	104	0.65	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
107	105	0.66	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
108	106	0.66	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
109	107	0.67	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
110	108	0.68	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
111	109	0.68	68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
112	110	0.69	115	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
113	111	0.69	204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
114	112	0.7	281	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
115	113	0.71	371	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
116	114	0.71	495	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
117	115	0.72	643	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
118	116	0.72	729	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
119	117	0.73	916	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120	118	0.74	932	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
121	119	0.74	1003	140	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
122	120	0.75	904	203	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
123	121	0.76	798	332	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
124	122	0.76	744	451	45	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
125	123	0.77	530	634	82	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
126	124	0.78	406	745	159	22	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
127	125	0.78	284	930	245	41	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
128	126	0.79	198	1034	359	59	11	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
129	127	0.79	121	1026	522	134	24	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0
130	128	0.8	81	995	709	200	57	15	2	1	0	0	0	0	0	0	0	0	0	0	0	0
131	129	0.81	44	908	886	361	109	39	11	2	0	0	0	0	0	0	0	0	0	0	0	0
132	130	0.81	22	742	982	543	209	67	29	7	2	0	0	0	0	0	0	0	0	0	0	0
133	131	0.82	13	622	1100	695	341	131	49	14	6	1	0	0	0	0	0	0	0	0	0	0
134	132	0.82	3	395	1098	832	469	246	86	39	8	2	1	0	0	0	0	0	0	0	0	0
135	133	0.83	2	272	984	1102	694	380	187	80	35	15	2	0	0	0	0	0	0	0	0	0

Table 6: Portions of the file generated by csuAnalyzePermute tool: The file Energy_PCA_MahCosine_HistCounts.txt containing the raw histogram counts of how often an algorithm recognized exactly *k* subjects at rank 1, 2, etc.

In the histogram count table, the third column labeled r1 indicates how many times out of 10,000 randomly generated probe and gallery sets the algorithm correctly recognized exactly rc subjects. So, looking at the upper left portion of the top table, we see that the PCA algorithm using the Mahalanobis Cosine distance exactly recognized 104 subjects six times in 10,000 trials. Scanning down the r1 column, the peak in this histogram is1003 at recognition count 119,

indicating that 119 subjects were recognized correctly more often than any other single number of subjects.

1	rc	rr	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10	r11	r12	r13	r14	r15
98	96	0.6	0.0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0
99	97	0.61	0.0002	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	98	0.61	0.0006	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101	99	0.62	0.0011	0	0	0	0	0	0	0	0	0	0	0	0	0	0
102	100	0.62	0.0017	0	0	0	0	0	0	0	0	0	0	0	0	0	0
103	101	0.63	0.003	0	0	0	0	0	0	0	0	0	0	0	0	0	0
104	102	0.64	0.0069	0	0	0	0	0	0	0	0	0	0	0	0	0	0
105	103	0.64	0.0112	0	0	0	0	0	0	0	0	0	0	0	0	0	0
106	104	0.65	0.0181	0	0	0	0	0	0	0	0	0	0	0	0	0	0
107	105	0.66	0.0272	0	0	0	0	0	0	0	0	0	0	0	0	0	0
108	106	0.66	0.037	0	0	0	0	0	0	0	0	0	0	0	0	0	0
109	107	0.67	0.0502	0.0002	0	0	0	0	0	0	0	0	0	0	0	0	0
110	108	0.68	0.0654	0.0005	0	0	0	0	0	0	0	0	0	0	0	0	0
111	109	0.68	0.0687	0.0014	0	0	0	0	0	0	0	0	0	0	0	0	0
112	110	0.69	0.0865	0.0026	0	0	0	0	0	0	0	0	0	0	0	0	0
113	111	0.69	0.0941	0.0045	0.0001	0	0	0	0	0	0	0	0	0	0	0	0
114	112	0.7	0.0921	0.0068	0.0003	0	0	0	0	0	0	0	0	0	0	0	0
115	113	0.71	0.0894	0.0121	0.0002	0	0	0	0	0	0	0	0	0	0	0	0
116	114	0.71	0.0829	0.0184	0.0002	0	0	0	0	0	0	0	0	0	0	0	0
117	115	0.72	0.0664	0.0284	0.0007	0	0	0	0	0	0	0	0	0	0	0	0
118	116	0.72	0.057	0.0425	0.0015	0.0002	0	0	0	0	0	0	0	0	0	0	0
119	117	0.73	0.0467	0.0569	0.0042	0.0004	0	0	0	0	0	0	0	0	0	0	0
120	118	0.74	0.0349	0.0723	0.0079	0.0003	0	0	0	0	0	0	0	0	0	0	0
121	119	0.74	0.0234	0.0797	0.01	0.0013	0	0	0	0	0	0	0	0	0	0	0
122	120	0.75	0.0136	0.0941	0.0201	0.0017	0.0002	0	0	0	0	0	0	0	0	0	0
123	121	0.76	0.0096	0.0952	0.0313	0.0036	0.0006	0.0001	0	0	0	0	0	0	0	0	0
124	122	0.76	0.0054	0.0959	0.0439	0.007	0.0012	0.0003	0	0	0	0	0	0	0	0	0
125	123	0.77	0.0035	0.0895	0.0613	0.0104	0.0026	0.0001	0.0001	0	0	0	0	0	0	0	0
126	124	0.78	0.0015	0.0822	0.0733	0.0188	0.0035	0.0007	0.0001	0	0	0	0	0	0	0	0
127	125	0.78	0.0008	0.0604	0.0871	0.0316	0.0069	0.0022	0.0003	0	0	0	0	0	0	0	0
128	126	0.79	0.0007	0.0543	0.1007	0.0499	0.0118	0.0024	0.0005	0.0001	0	0	0	0	0	0	0
129	127	0.79	0	0.0375	0.1014	0.061	0.0217	0.0054	0.0011	0.0002	0.0001	0	0	0	0	0	0
130	128	0.8	0	0.0261	0.1004	0.0828	0.0348	0.0107	0.003	0.0014	0.0002	0	0	0	0	0	0
131	129	0.81	0	0.0173	0.0912	0.0903	0.0543	0.0204	0.0061	0.0022	0.0004	0.0001	0	0	0	0	0
132	130	0.81	0	0.0093	0.0779	0.105	0.0638	0.0309	0.0137	0.0034	0.001	0.0004	0.0001	0	0	0	0
133	131	0.82	0	0.0057	0.0626	0.1073	0.0827	0.0497	0.0209	0.0087	0.0033	0.0007	0.0003	0.0001	0	0	0
134	132	0.82	0	0.0031	0.0437	0.1025	0.101	0.0629	0.0307	0.016	0.0073	0.0031	0.0013	0.0003	0.0001	0	0
135	133	0.83	0	0.0016	0.0325	0.0879	0.1108	0.0819	0.051	0.0247	0.0109	0.0048	0.002	0.001	0.0004	0.0001	0

Table 7: Portions of the file generated by csuAnalyzePermute tool: the file Energy_PCA_MahCosine_HistProbs.txt containing the sample probability distribution derived from the histogram obtained by dividing through by the number of random trials: 10,000 in this case.

In Table 7, the results are obtained by dividing the histogram counts by the number of trials (10,000). However, the interpretation is now very elegant, each column starting with r1 is a sample probability distribution for recognition

count, or equivalently recognition rate. Column r1 corresponds to recognition rank 1, r2 to recognition rank 2, etc.

To compute the average recognition rate at each rank [29], and the mode of the distribution at each rank is now a simple process. Perhaps more importantly, it is also possible to compute a two-sided confidence interval. This is done by simply identifying the central portion of each distribution accounting for 95 percent of the outcomes, i.e., finding the lower and upper bounds on the confidence interval. The algorithm starts at rc equal to zero and sums the values in the histogram until reaching 250. The cutoff of 250 is 2.5 percent of 10,000. The analogous operation is performed coming down from the top to compute the upper bound. The third file produced by csuAnalyzePermute gives the average, mode and upper and lower confidence bounds on recognition count for rank 1, 2, etc. It is essentially a cumulative match curve with 95 percent confidence error bars. Table 8 shows the first 10 lines of the file Energy_PCA_MahCosine_CMCurve.txt.

rank	lower	mode	upper	mean
1	110	119	127	118.6
2	119	126	134	126.8
3	124	131	139	131.3
4	127	135	141	134.4
5	130	138	143	136.5
6	131	138	144	138.1
7	133	139	146	139.4
8	134	141	147	140.6
9	136	141	148	141.8
10	137	143	149	142.8

Table 8: Above table shows top 10 lines of the Energy_PCA_MahCosine_CMCurve.txt file, providing a cumulative match curve with error bars.

The first row contains column headers, and the columns are: the recognition rank, the lower bound on the 95 percent confidence interval, the mode of the probability distribution, the upper bound on the confidence interval and the mean recognition count. From these files, it is relatively simple to generate cumulative match curves with associated error bars. However, do note that this file records recognition counts, and all the values must be divided by the number of probe images to arrive at recognition rates.

4.1.3 Data Set:

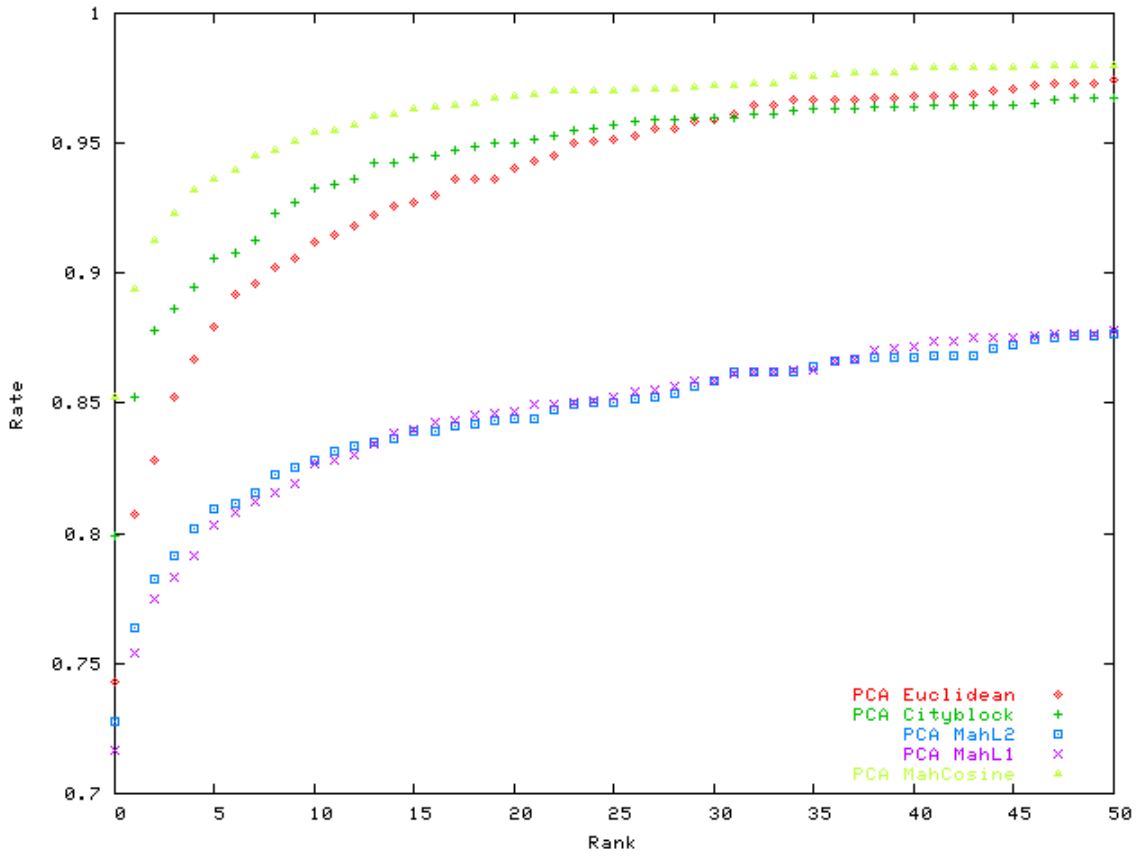
The recognition algorithms are tested using a set of Probe Images, P, and a set of Gallery images, G. In all the experiments, the Gallery images were obtained from the FA gallery set (FA ε G), while the Probe images were obtained from FB, FC, Duplicate I, or Duplicate II ({FB,FC,Duplicate I, Duplicate II} ε P). For the fifth experiment, we also conducted a test that involved the probe and gallery images that were drawn from the 160 people for whom there are four or more images. Every person of this 160 people had two images per day, such that the images captured two distinct expressions, and the images were taken for two different days generating four images per person, equivalently 640 images for 160 people. This resulting 640 test images were used to generate probe and gallery set by random permutation.

We determined the face identification performance using the PCA technique under different distance measures such as L1 (city block), L2 (Euclidean distance), Mahalanobis cosine angle, Mahalanobis L1, and Mahalanobis L2. In experiment I, we obtained the results that agree with our previous conclusion,

specifically, Mahalanobis Cosine distance measure gives best results with PCA face recognition system followed by L1 (city block) measure. We compared the cut-off methods that provided the best face identification results, in the experiment II. The available cut-off methods were: Simple, Energy, Stretch, and None. In the experiment III we studied the effect of lossy and lossless eigenvector elimination. In the experiment IV, the EBGM algorithm was compared with results obtained using the two different training wavelets, specifically, Gabor Wiskott, and Gabor Bolme wavelets. After ascertaining the Mahalanobis cosine distance as the best measure with the Energy cut-off PCA technique, we compared the PCA technique to Elastic Bunch Graph Matching technique in Experiment V.

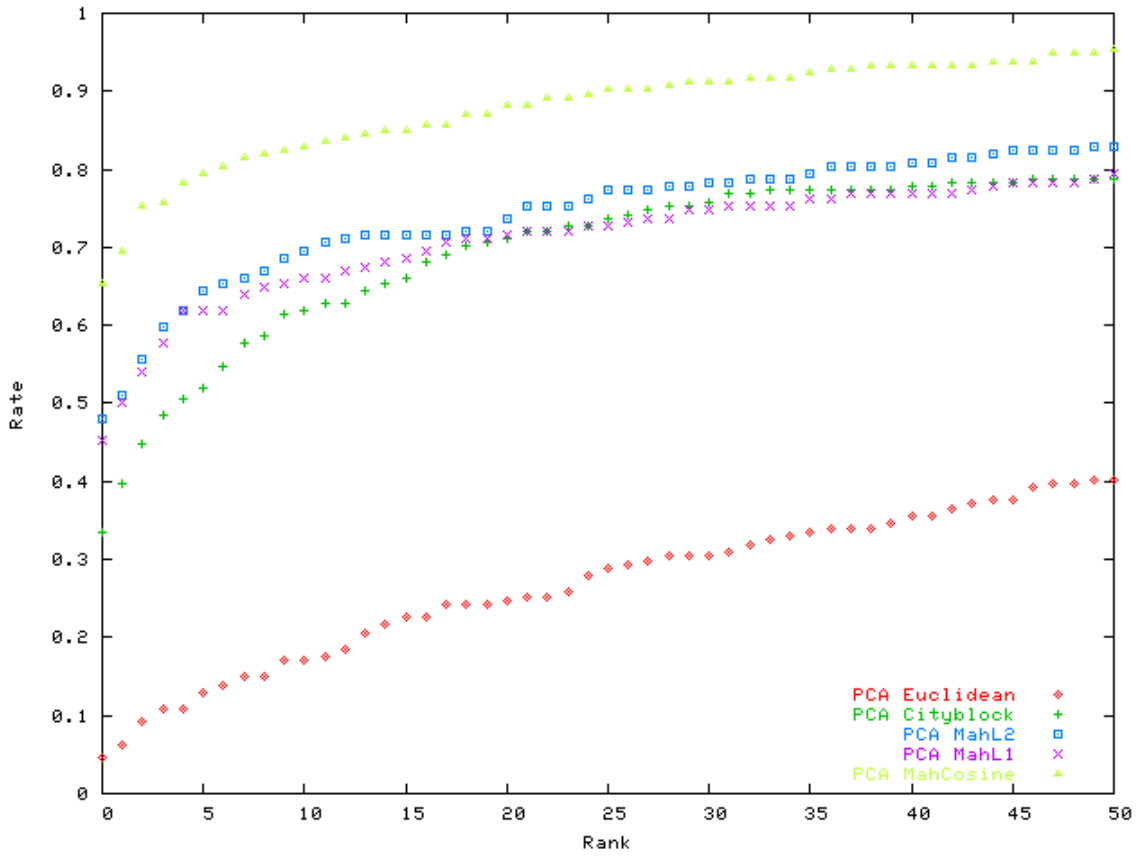
4.2 Experiment I: PCA performance effect due to selection of different distance measures

Having studied the results obtained from [3], we can be certain that any combination of the standard distance measures (L1, L2, Mahalanobis cosine and Mahalanobis angle) is not significantly better than the individual distance measures.



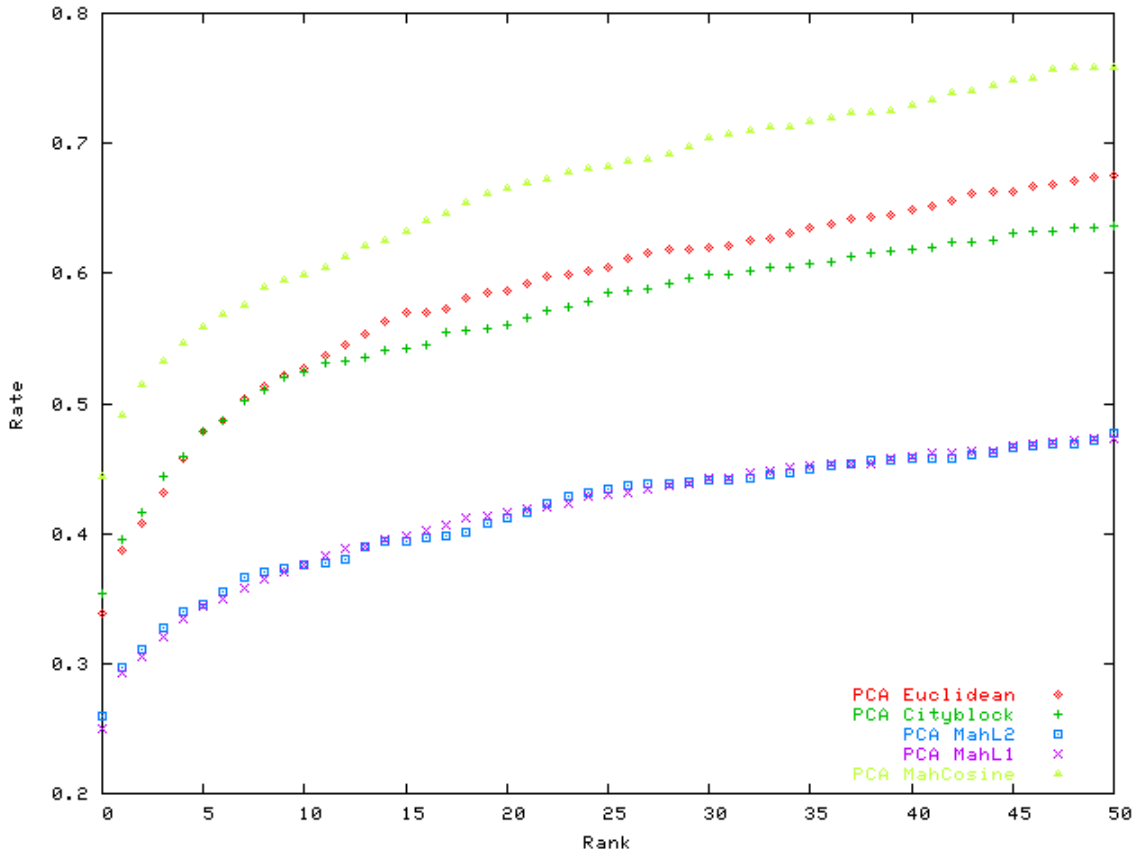
(a)

In experiment I, the training set was held constant and five different distance measures: city block (L1), Euclidean distance (L2), Mahalanobis cosine, Mahalanobis L1, and Mahalanobis L2 were compared in determining the face recognition rate with the PCA technique using Simple cutoff criteria.



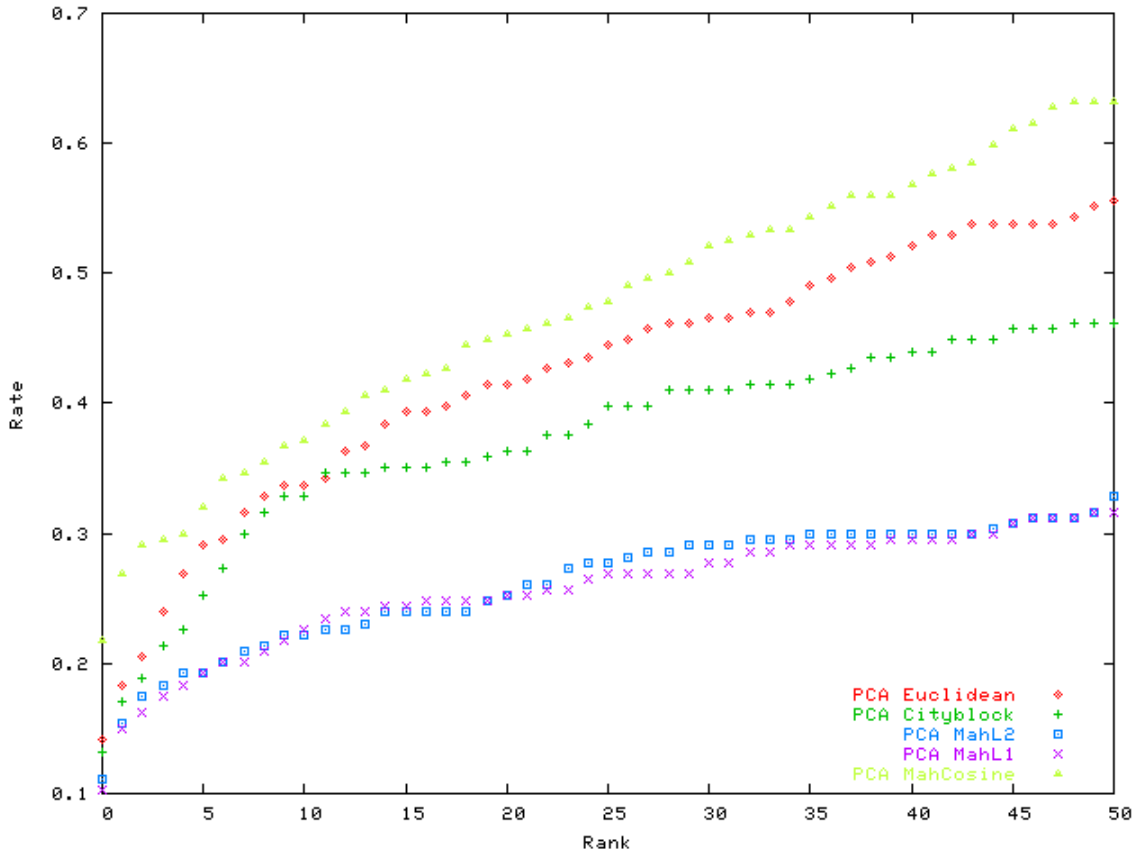
(b)

The experiments were conducted on the FB, FC, duplicate I, and duplicate II categories. The results are shown below in Figure 1. As can be seen from Figure 1a, Mahalanobis cosine distance measure generates the best recognition percentage rate at different ranks followed by city block measure.



(c)

Similarly, for the Figure 1b we clearly observe that Euclidean is worst distance measure and Mahalanobis cosine is the best distance measure. In Figure 1c, we probed DUPLICATE I set in the FA gallery and we see that Mahalanobis cosine distance measure still generates the best recognition rate. DUPLICATE II probe set was searched in the FA gallery set and the resulting recognition rate is shown in Figure 1d.

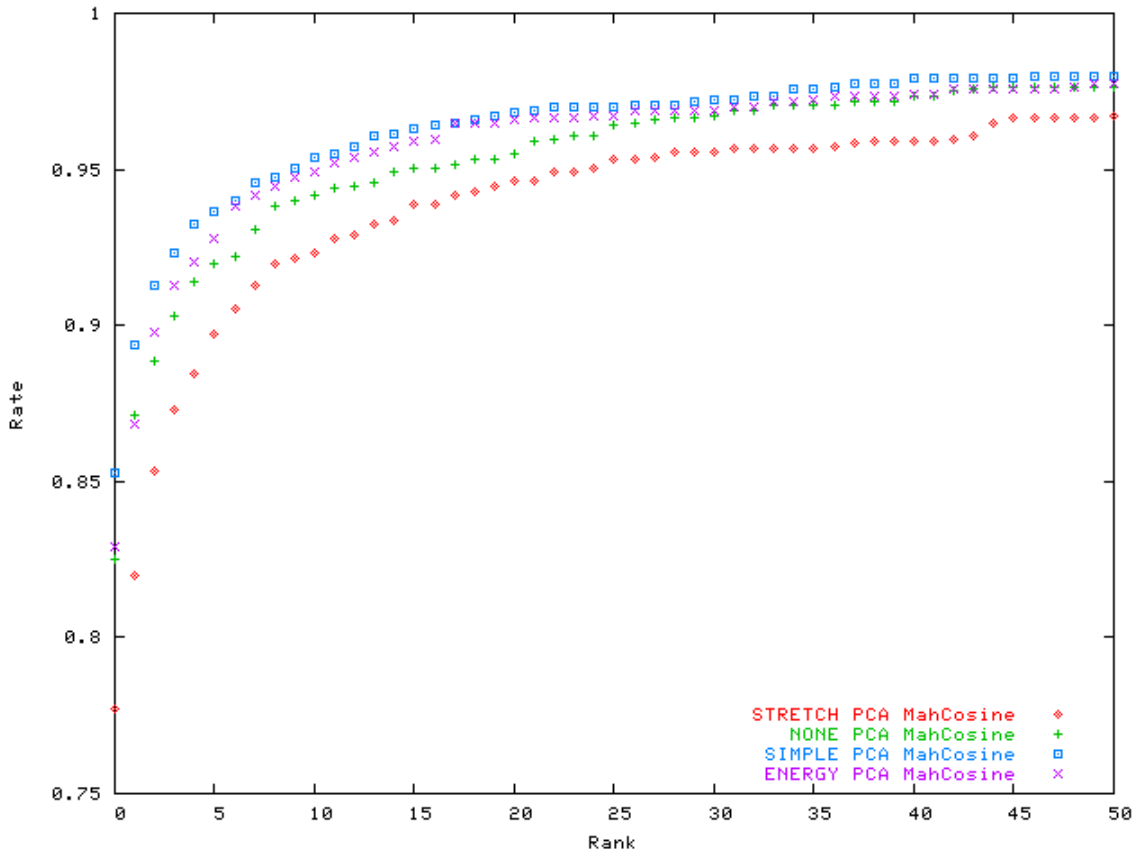


(d)

Figure 1: Cumulative Match Curves: a) FERET FB Probe Set. b) FERET FC Probe Set. c) FERET Duplicate I Probe Set. d) FERET Duplicate II Probe Set.

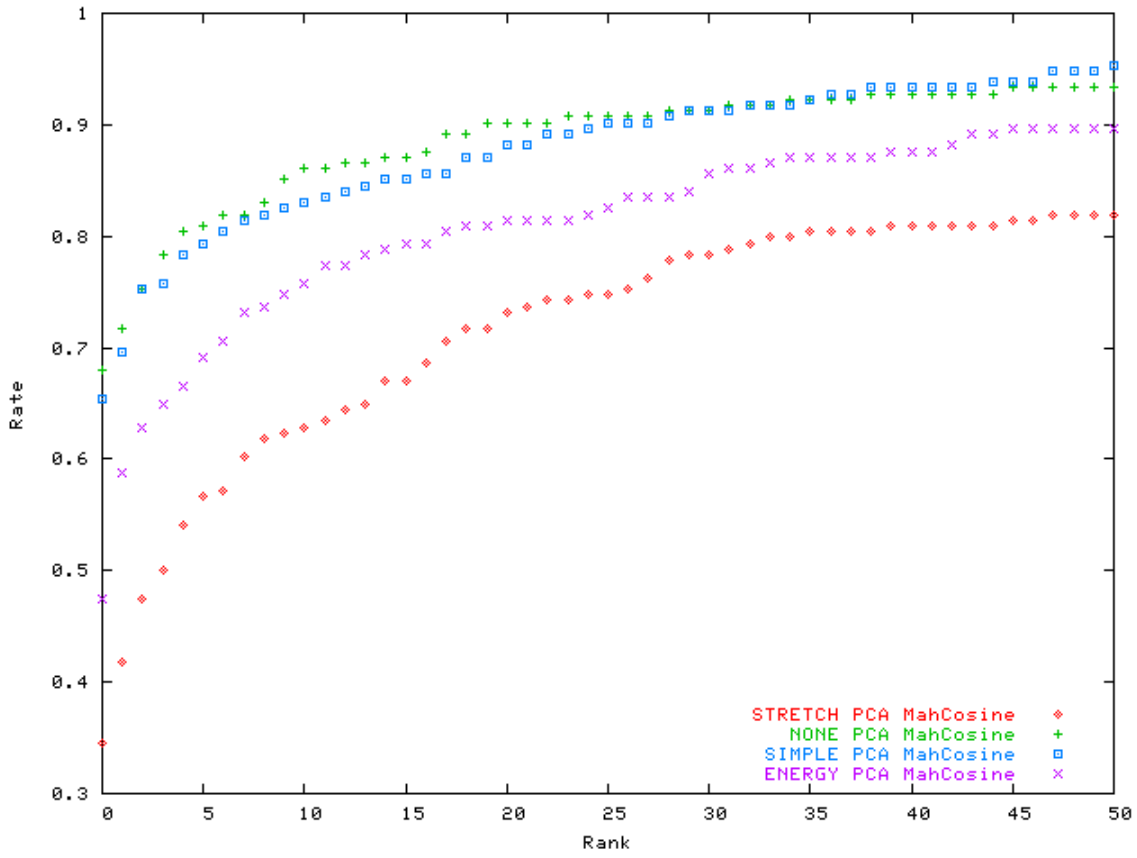
4.3 Experiment II: Comparing available Eigenvector cutoff methods in PCA algorithm.

In experiment II we studied the PCA algorithms by varying cutoff methods that retain only critical Eigenvectors. The CSU system provides four different cut-off methods to retain critical eigenvectors. The cut-off methods are None, Simple, Energy, and Stretch. None cut-off method retains all the eigenvectors, while the Simple cut-off method retains a percentage of the eigenvectors and defaults to 60 percent.



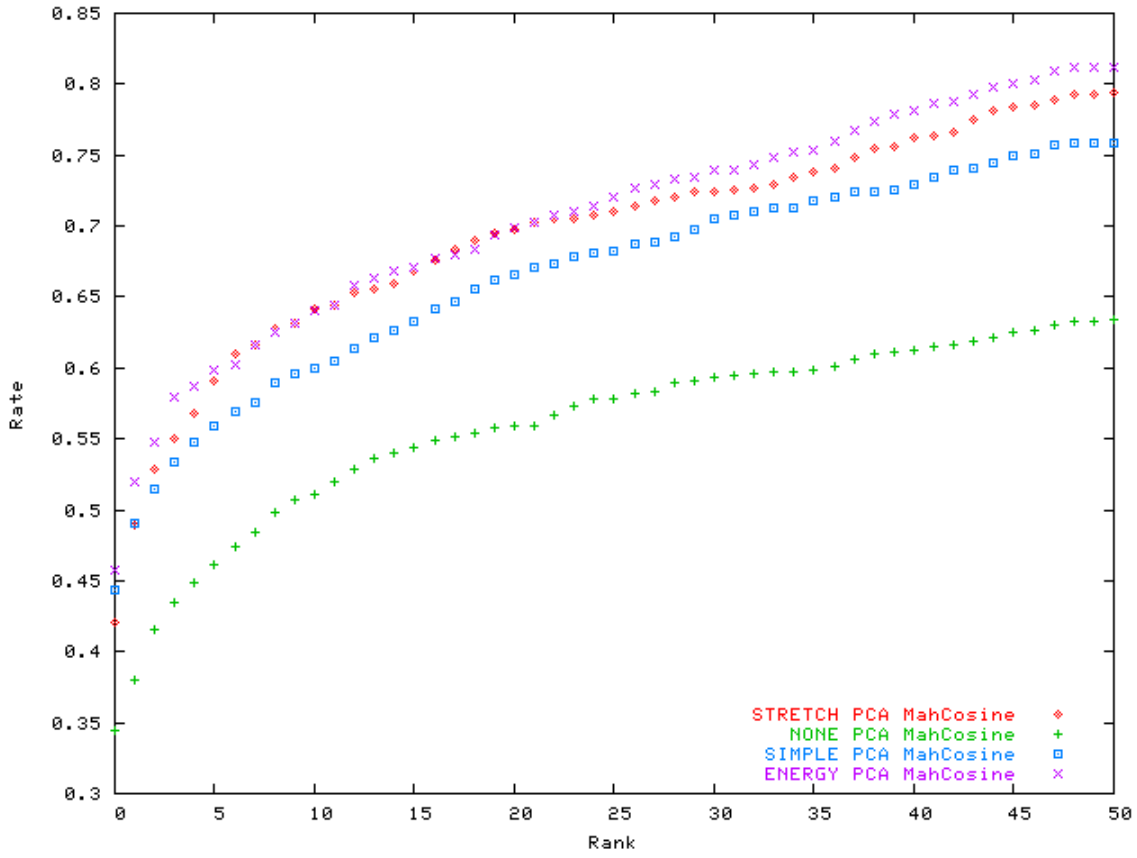
(a)

The Energy cut-off method retains eigenvectors accounting for a percentage of the total energy and defaults to 90 percent. The Stretch cut-off method retains all the eigenvectors greater than a percentage of the largest eigenvector. The results obtained are given below in Figure 2.



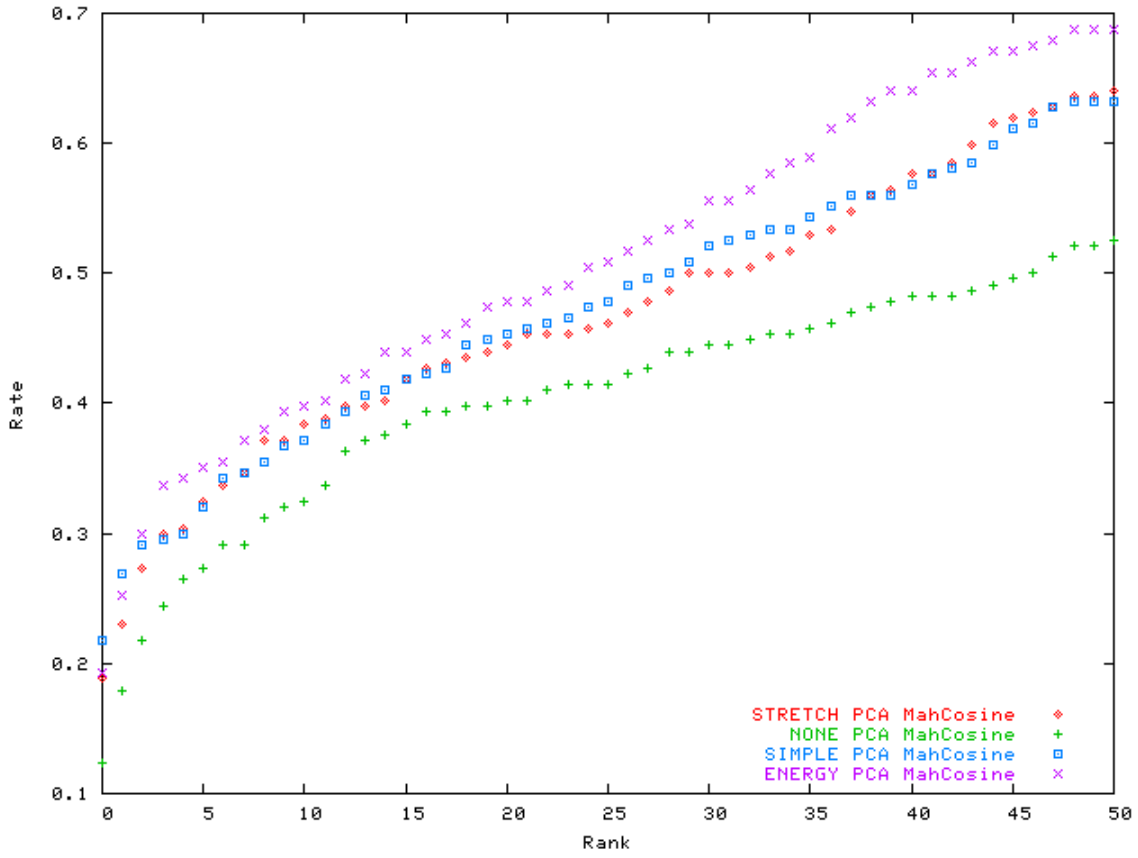
(b)

As defined in the FERET system, the FA represents the gallery set. The FB and FC sets represent the image of the subject taken at the same time but with different expression and under different lighting condition respectively. While, Duplicate I and Duplicate II represent the duplicates. Identification using the duplicate probe set is most important and a difficult task in face identification. Therefore, the identification task represents the solution to the elementary problem in successful implementation of real-life face identification system.



(c)

By using cutoff methods we represent the critical information in a compressed format and thereby reduce the comparison time. Looking at the Figure 2.a, we observe that the system obtains best results using SIMPLE cutoff method that retains only 40 percent of top eigenvectors. The FC FERET set represent pictures taken of the subject under different lighting conditions. For FC probing, the system obtained best results by retaining all the eigenvectors ascertaining the fact



(d)

Figure 2: Comparison of Cutoff methods in the PCA technique: a) FERET FB probe gave best results by retaining only 60 percent of the top eigenvectors. b) FERET FC probe gave best result by retaining all the eigenvectors and similar result using SIMPLE technique. c) FERET Duplicate I probe generated best result by retaining the eigenvectors containing 90 percent of total energy. d) FERET Duplicate II probe also performed best using Energy cutoff method.

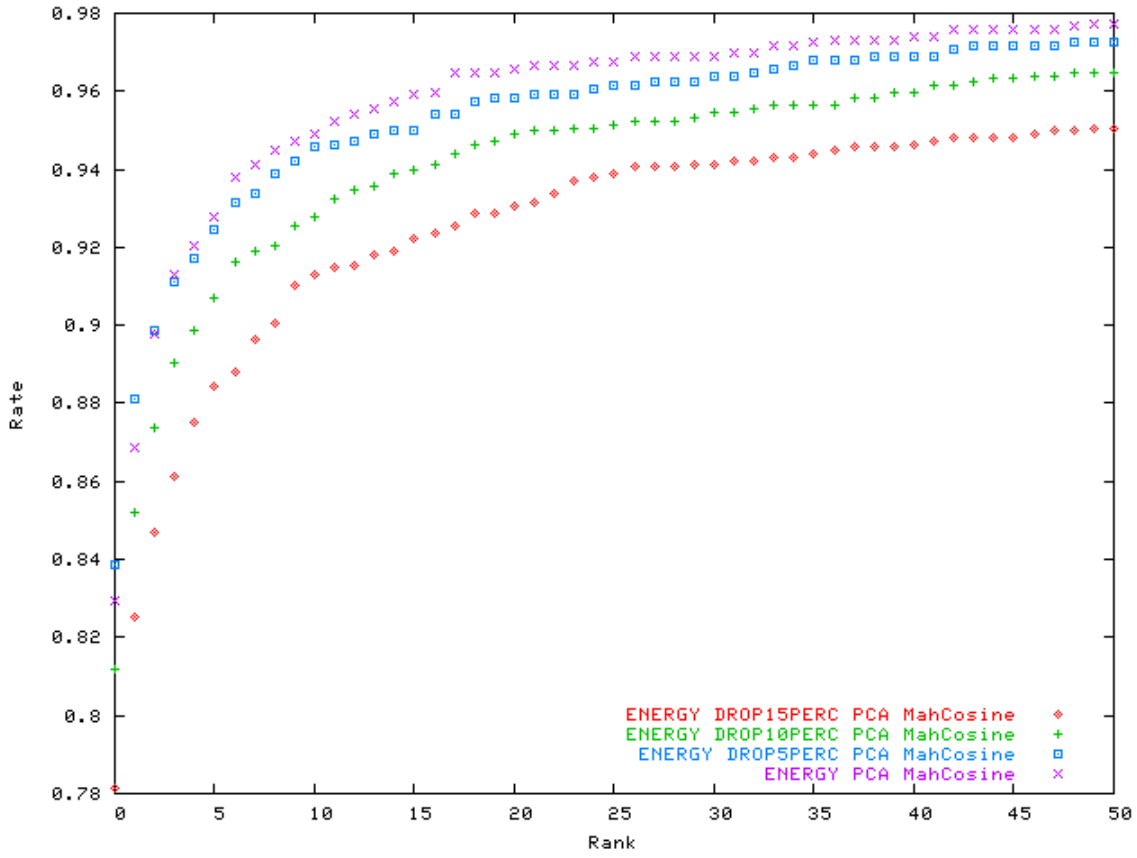
that PCA technique is sensitive to significant feature changes and external conditions such as lightning, occlusion, etc. In general, for FB and FC probe the SIMPLE cutoff method generated better result in comparison to other available methods, excluding the NONE option for FC.

For Duplicate I, and Duplicate II probe set the system generated best percentage match using Energy method. Though, results obtained with FB, and FC probe set are important, but DUPLICATE I, and DUPLICATE II represent

difficult probe set in face recognition. Duplicate probe sets represent real life scenarios and hence, results obtained in Figure 2.c, and Figure 2.d are more important for us. Energy cutoff method generates best result for the duplicate probe sets and so we compare Energy cutoff PCA technique to the EBGM algorithm.

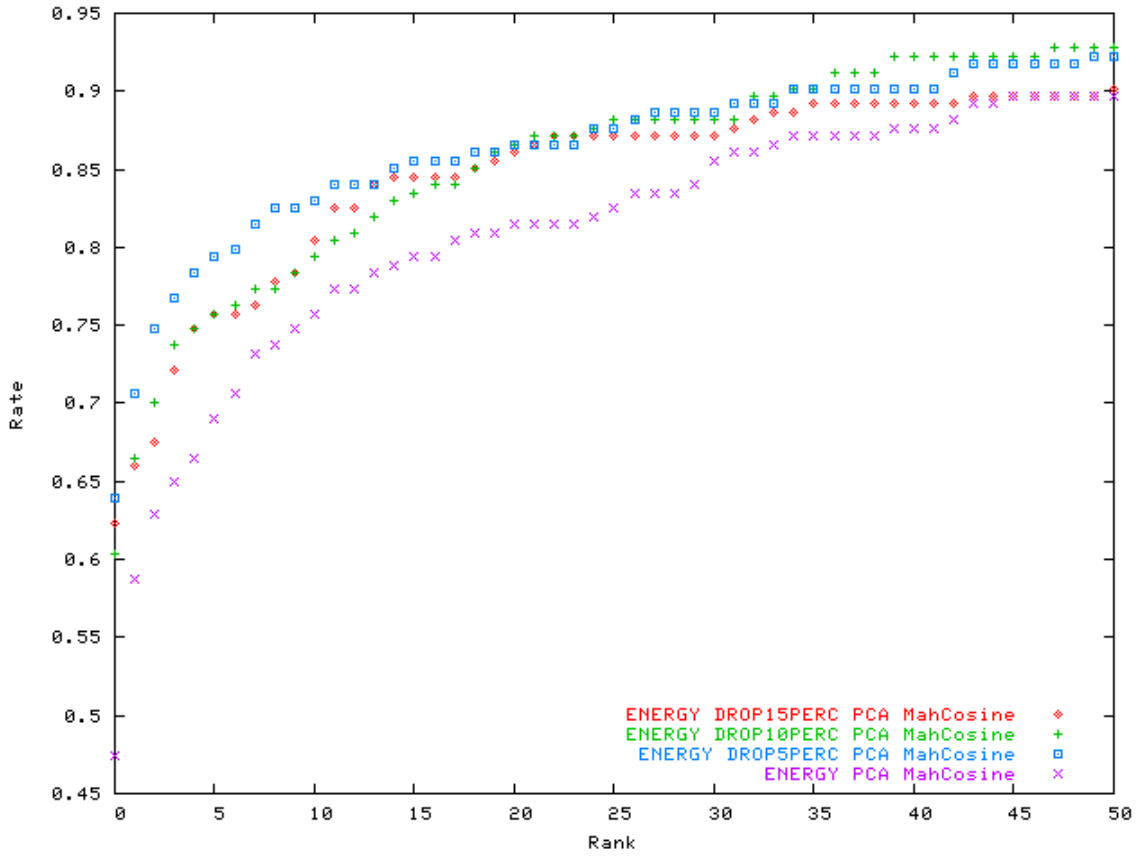
4.4 Experiment III: Comparing the effect of dropping top Eigenvectors before training PCA.

Remember from the PCA discussion that each Eigenvector captures a particular type of feature from the given image. Some eigenvectors capture lightning, illumination, and similar non-facial features. While other Eigenvectors capture facial expression features. In fact, we can even determine the relationship between human face features and the given Eigenvectors, to a certain extent, by looking at the generated Eigenvectors. Hence, by carefully dropping particular eigenvectors the system can discard the features that generate erroneous results. In the present experiment, we studied the difference in recognition by varying the Eigenvector count on face identification. From Experiment II we concluded that PCA technique with Energy cutoff generated best results, hence we used this technique for further analysis.



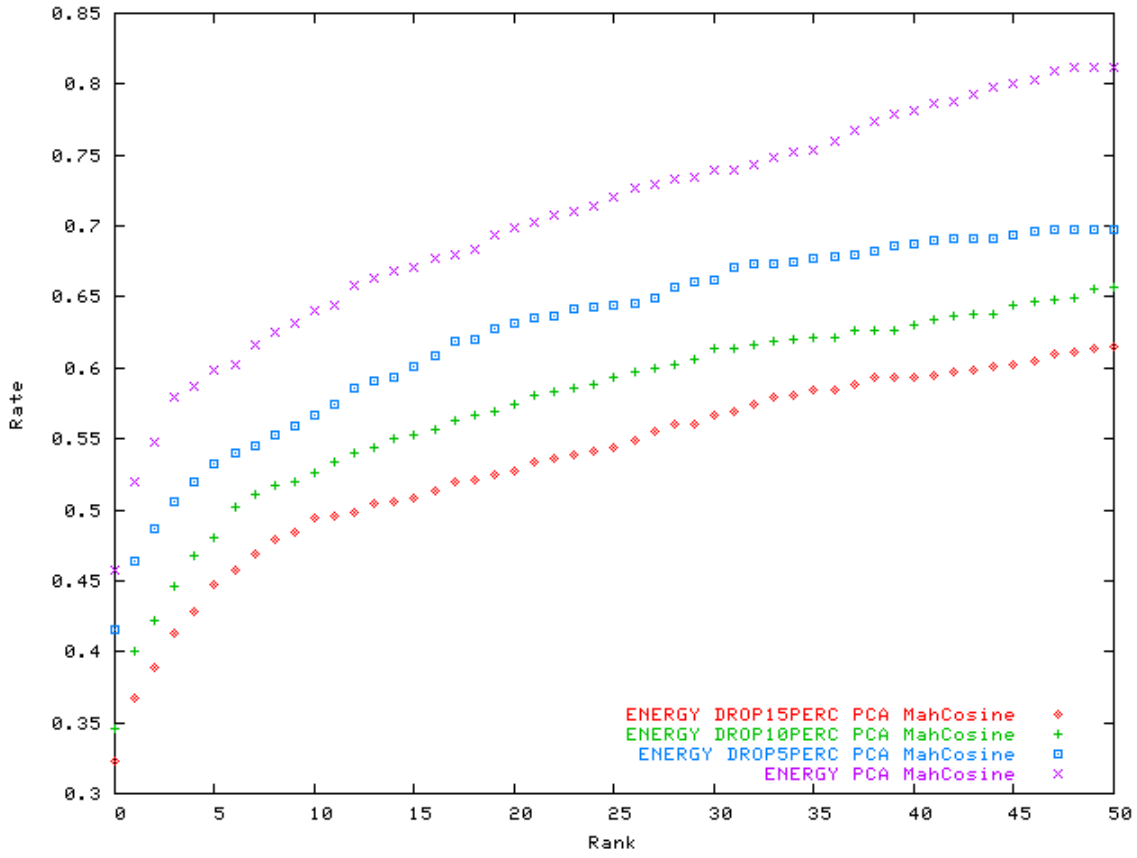
(a)

In the current experiment we determined the results using the PCA algorithm with Energy cutoff particularly, dropping the top 5 percent, 10 percent, or 15 percent eigenvectors and further retaining only those eigenvectors that capture 90 percent of total energy.



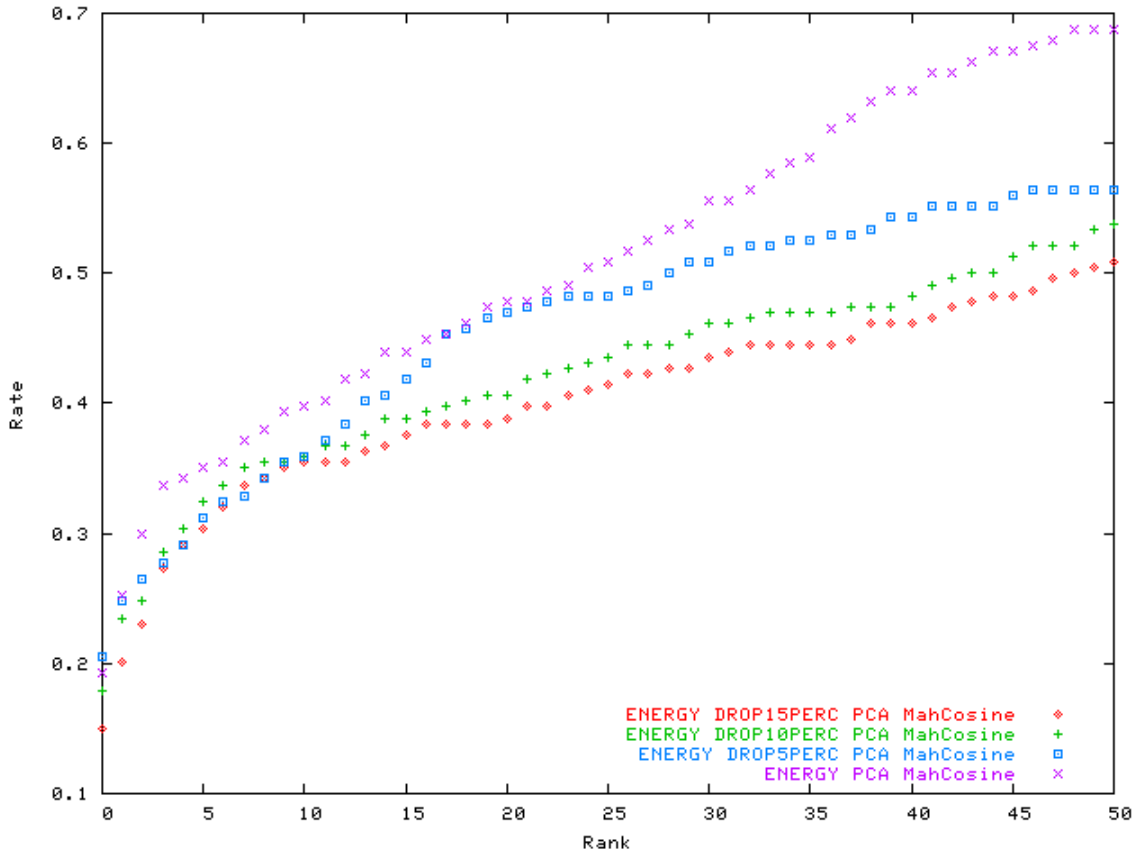
(b)

Since the CSU system does not provide a method to drop intermediate eigenvectors, our experiment concentrated on dropping the top eigenvectors. After the eigenvectors were dropped, we obtained the subset of Eigenvectors that was used as a sub-space into which comparison was performed. Intuitively, by removing the appropriate Eigenvectors, we excluded those eigenvectors that either re-capture similar data or capture unimportant non-facial data, thereby producing better results.



(c)

Additionally, the system performance improved due to reduction of processing steps and possible exclusion of similar data. By selecting a proper subset of Eigenvectors we improved the real-time performance using the PCA algorithm. Results are shown below in Figure 3.



(d)

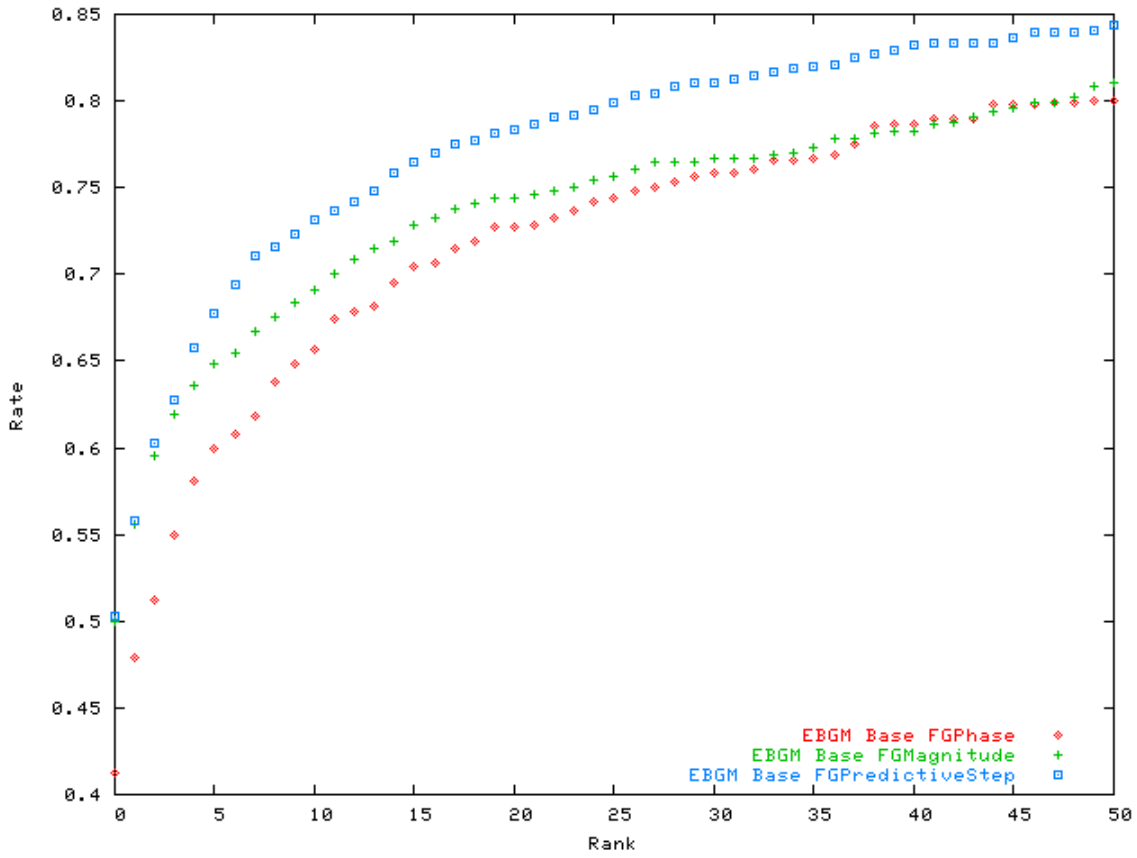
Figure 3: Difference in recognition rate after dropping 0%, 5%, 10%, and 15% eigenvectors to train PCA algorithm. a) For FERET FB probe the system gives best recognition rate close to 98% by using all the eigenvectors. b) In FERET FC probe the system gives better recognition between 5%-10% dropped eigenvectors, but as this percentage increases further the recognition rate drops. c) FERET Duplicate I probe generated best result by retaining all the eigenvectors containing 90% of total energy. d) FERET Duplicate II probe also performed best without dropping any eigenvectors with Energy cutoff method.

The results presented in figure 3 suggest that the face identification system gave superior results when all the eigenvectors were used for training the PCA system. Only exception was the results obtained for FC probe, the FC probing gave best result by dropping 5-10 percent eigenvectors. But, the recognition dropped after dropping more than 10 percent eigenvectors or by retaining all the

eigenvectors. For FERET duplicate I, and II sets the recognition rate dropped, but by retaining all the eigenvectors we obtained better results.

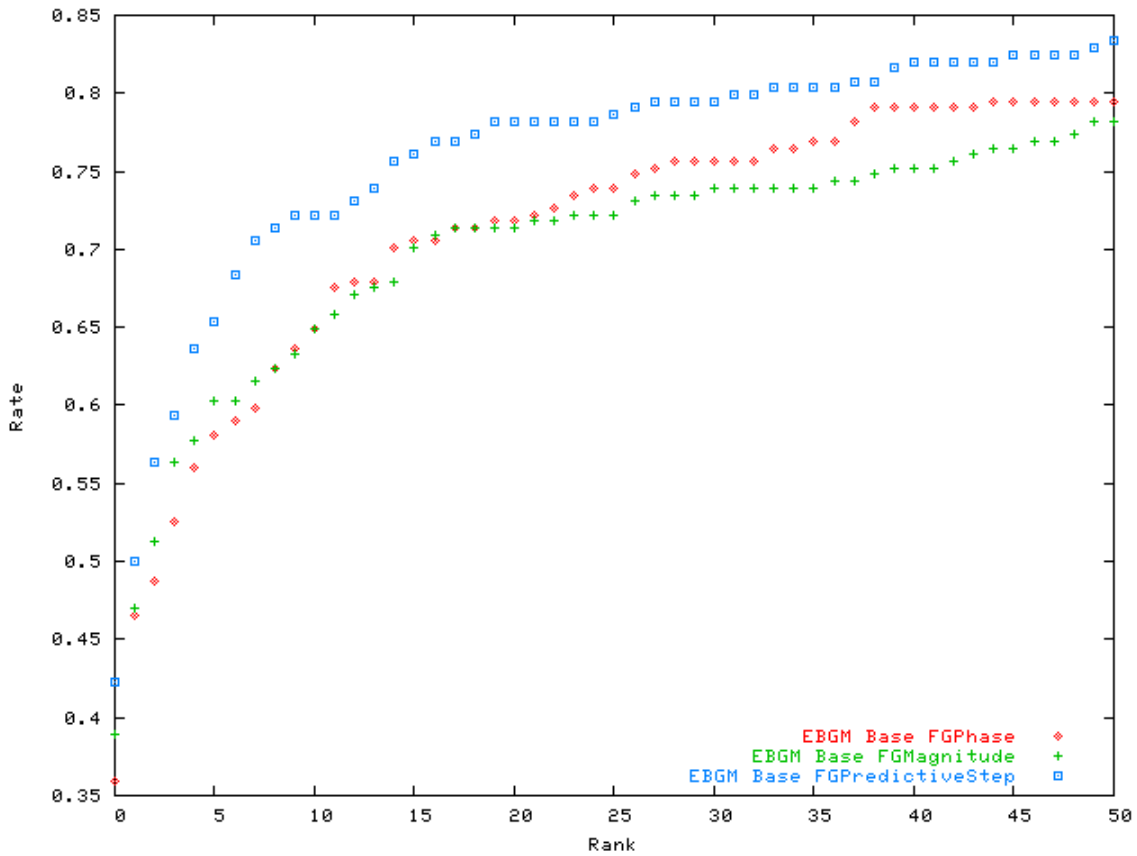
4.5 Experiment IV: Comparing similarity distance measure for EBGM algorithm.

In experiment IV, we compared the results obtained by training the EBGM algorithm using: Bolme and Wiskott wavelets on Duplicate I and Duplicate II probe set. The Bolme wavelet was used in the CSU version of the EBGM algorithm, where as Wiskott wavelet was used by Boschum in [29]. We used the Bolme wavelet to train EBGM algorithm and fitting graph with the predictive iteration method and predictive step measurement as described in [29].



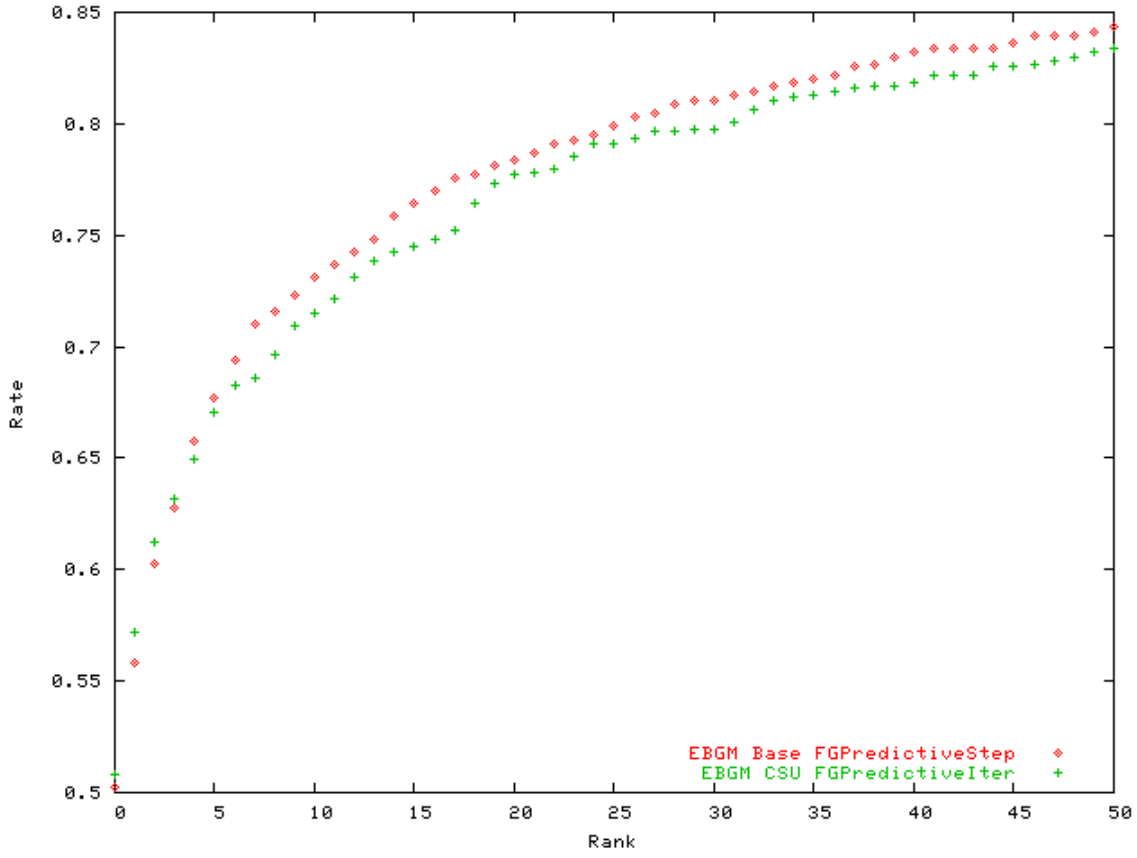
(a)

While in the rest of the current experiment we used predictive step to fit the model graph to novel images. We then compared the similarity distance measure with the Wiskott wavelet using magnitude, phase, and predictive step distance measures. Once we found the similarity measure that generated best recognition rate in conjunction with the Wiskott wavelet, we compared this similarity measure with the Bolme wavelet and predictive step distance measure.



(b)

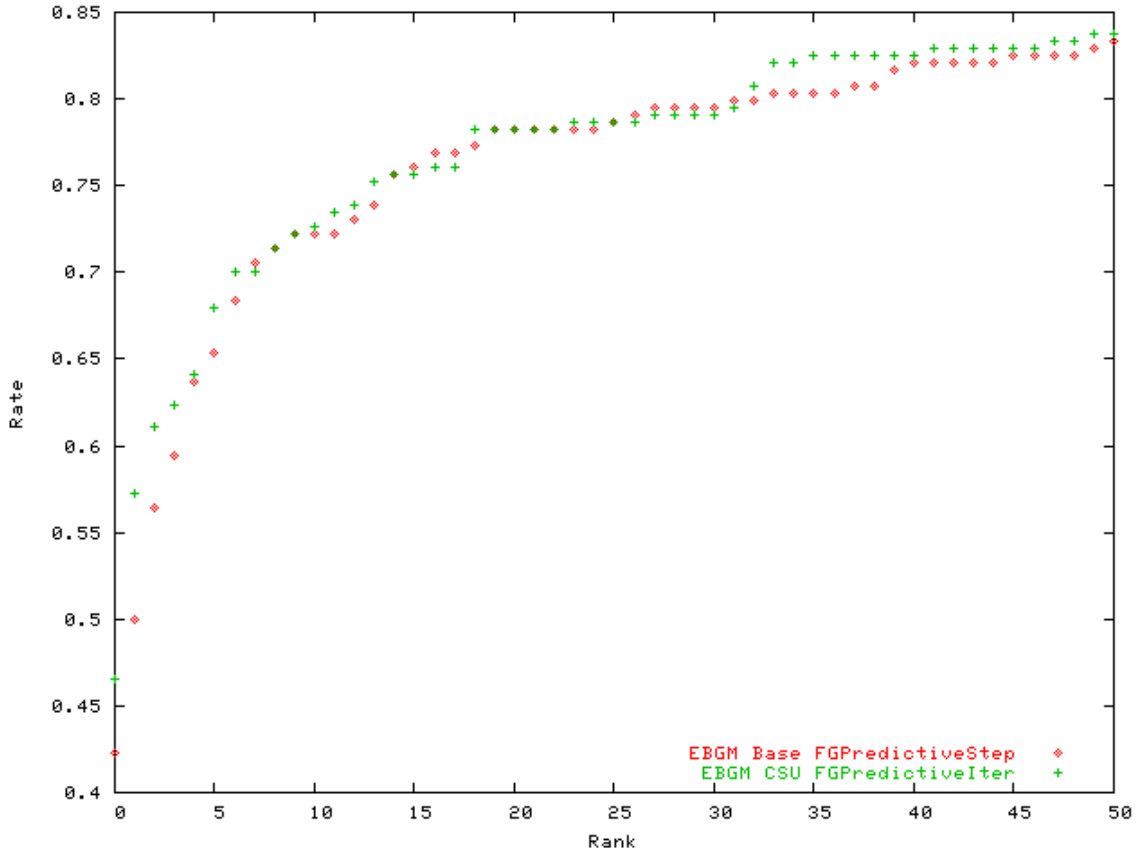
The EBGM algorithm is extremely CPU demanding algorithm. Compared to the PCA algorithm, the EBGM algorithm ran for seven hours in the simple similarity distance measure, compared to less than an hour for PCA. For complex distance measure such as the grid sample, and the fixed local search, the process expected to run over 1023 hours and 93 hours respectively. Such a long time period is a huge disadvantage for implementing a real-life face identification system, and, therefore, the distance



(c)

measures that expected to run over eight hours were discarded from further discussion and analysis.

The results obtained for the experiment IV are shown in Figure 4. In figure 4.a, we compared phase, magnitude, and predictive step similarity measures with the Wiskott wavelet on the Duplicate I probe set. The predictive step similarity measure generated best results for the Duplicate I probe. We performed a similar experiment on the Duplicate II probe set, which also generated equivalent results. In rest of the experiment, we maintained the same similarity measure, Predictive step, to compare the wavelet and graph fitting effects.



(d)

Figure 4: Comparison of wavelet used in generating model graph and use of different distance similarity measure: a) Comparison of different distance measures on Duplicate I probe with Wiskott wavelet gave best result using Predictive step similarity measure. b) On Duplicate II probe set with Wiskott wavelet also generated best result using Predictive step similarity measure. c) Duplicate I probe set with Wiskott wavelet and predictive step measure

performs better than Bolme wavelet. d) Recognition rate for Duplicate II probe set generated equivalent result using either Wiskott wavelet or Bolme wavelet.

Figure 4.c, presents a comparison of the Wiskott wavelet with Predictive step graph fitting and the Bolme wavelet with Predictive iteration graph fitting on Duplicate I. For the Duplicate II probes, Wiskott and Bolme wavelets generated almost similar results. The result of performing permutation analysis on the Wiskott, and Bolme wavelets technique is given in table 9.

Algorithm 1	Algorithm 2	Mean	Mode	Lower	Upper	P(Alg1 > Alg2)	P(Alg1 < Alg2)
EBGM_Base_FGPredictiveStep	EBGM_CSU_FGPredictiveIter	-0.35	-1	-8	7	0.4134	0.4844

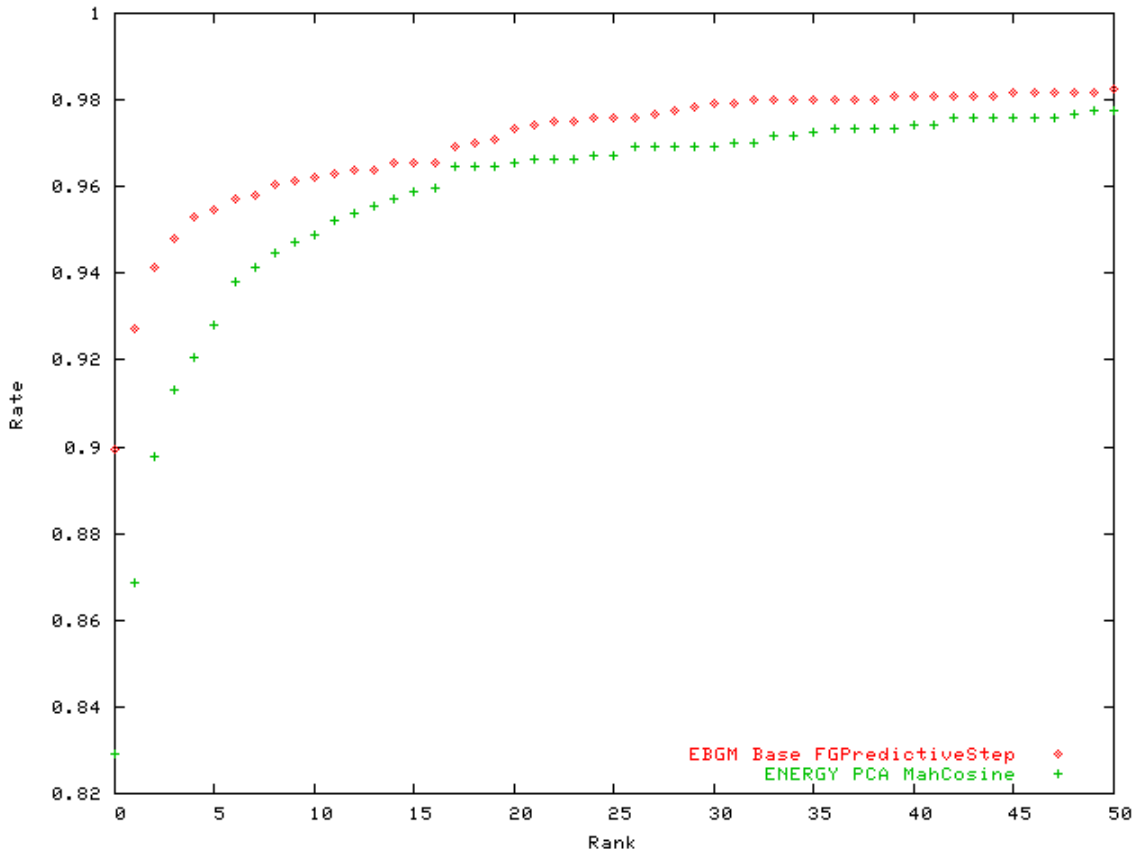
Table 9: Permutation analysis of Bolme and Wiskott wavelets.

We observe that neither wavelet usage affects the algorithm significantly.

4.6 Experiment V: Comparison of the PCA and the EBGM algorithm.

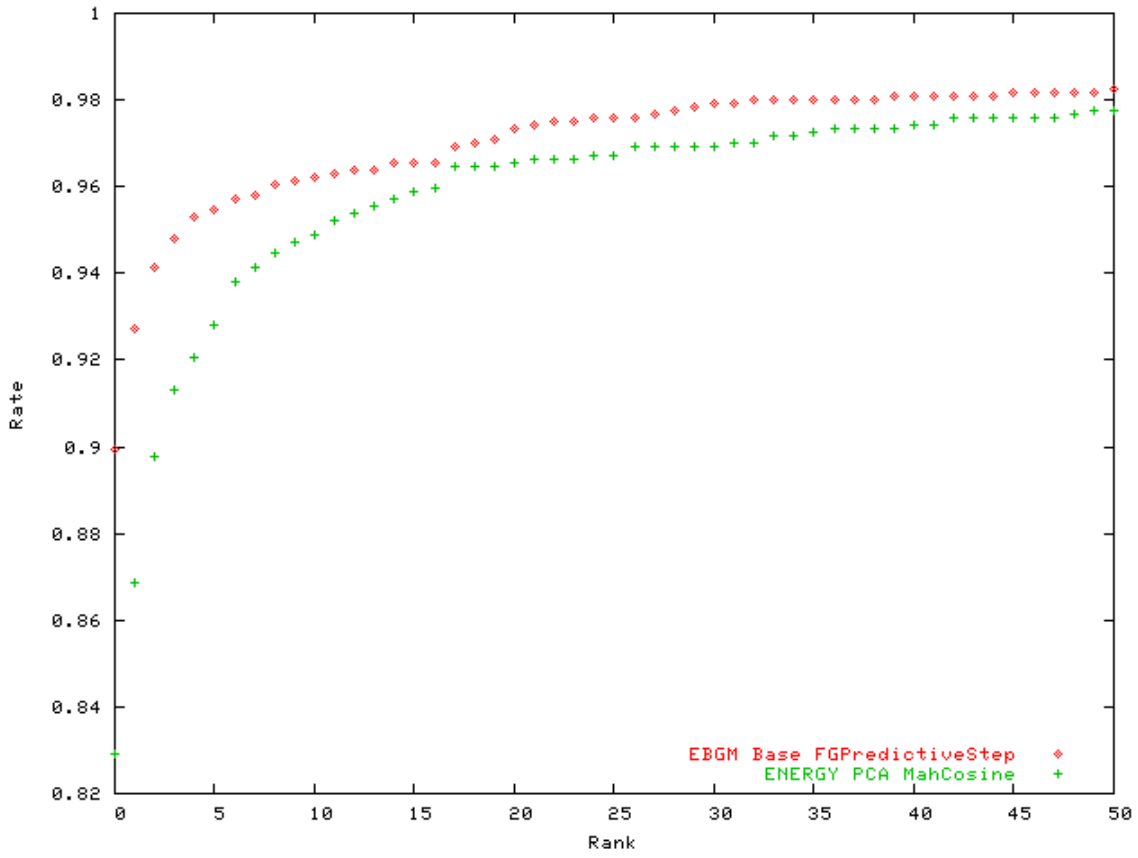
In the current experiment, we compared the PCA technique to the EBGM algorithm. The PCA algorithm was executed using the Mahalanobis cosine distance measure and lossless Energy cutoff method. Just to remind the reader, the Energy cutoff method retains only those critical eigenvectors that contain 90 percent of the total energy. The EBGM algorithm in experiment V used Wiskott wavelet with predictive step to generate the model graph from training images and used the predictive distance similarity measure.

The results obtained are shown below in Figure 5.



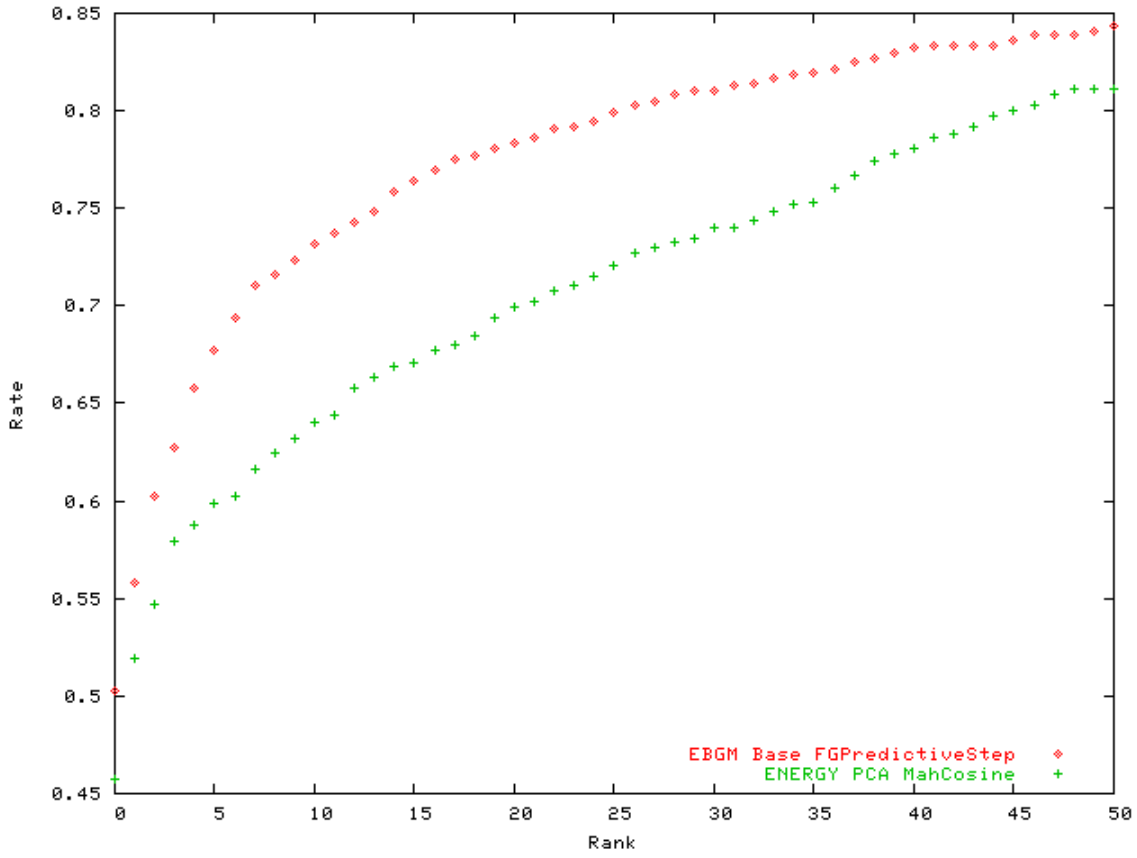
(a)

Looking at the results presented in the figure 5, we conclude that the EBGM algorithm gives better results overall. It must be noted that the EBGM algorithm produces substandard results at rank 0 in comparison to the PCA technique, but as the rank increases EBGM gives better recognition rate. After comparing the recognition rate between the PCA and the EBGM technique on the same probe and gallery sets, we then determined the comparison results using randomly generated probe and gallery sets.



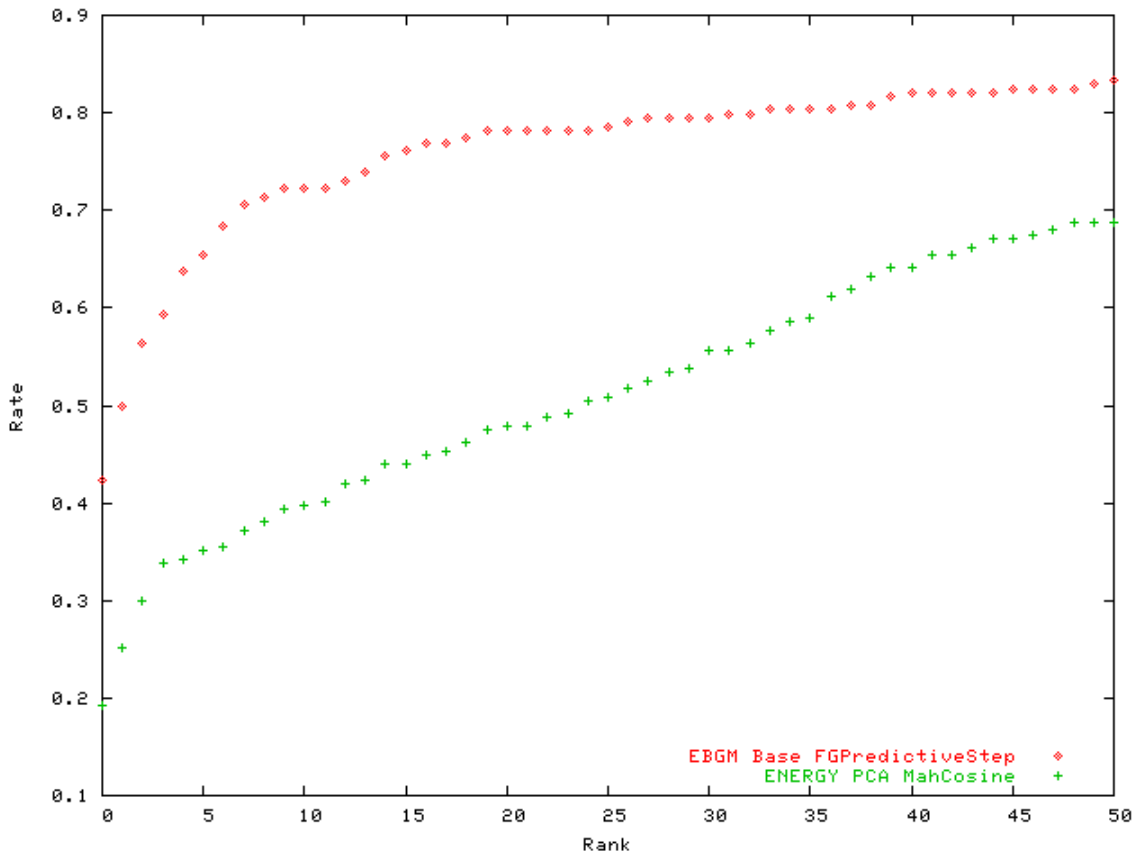
(b)

We used the permutation and analysis tool that randomly generates the probe and gallery sets. This randomly generated set was later used to find the best recognition rate at rank zero.



(c)

The probe and gallery set generation and recognition task were performed 10,000 times. However, as J. Ross et al. discussed [7], it must be noted that error bars overlap is not necessarily an appropriate test. Better is to compute the actual difference in recognition rates for each of the 10,000 trials, and then look at the distribution of this new random variable. With this test, interpreting the resulting distribution is relatively easy. More positive values indicates one algorithm is better, more negative values indicates the other algorithm is better, and distributions centered around zero indicate there is no difference between the algorithms.



(d)

Figure 5: Comparison of the PCA and the EBGM algorithm: a) FB probe set gives better recognition rate using EBGM algorithm. b) FC probe also generate better results with EBGM algorithm. c) Duplicate I recognition rate comparison between the EBGM and the PCA technique. d) Duplicate II recognition rate comparison between the EBGM and the PCA technique. e) Random permutation of gallery and probe set over 10,000 times and average recognition rate comparison between the EBGM and the PCA technique.

Table 10 shows the permutation statistics that were observed while comparing the PCA and the EBGM algorithm. Table 11 presents pair-wise comparisons based upon the difference in the recognition rate statistic.

Permutation Results:

Algorithm	Mean	Mode	Lower	Upper	Mean%	Mode%	Lower%	Upper%
EBGM_Base_FGPredictiveStep	111.8	111	104	120	69.9%	69.4%	65.0%	75.0%
ENERGY_PCA_MahCosine	118.6	119	110	127	74.1%	74.4%	68.8%	79.4%

Table 10: Permutation analysis results between the PCA and the EBGM algorithm.

Comparing Algorithms (Alg1 - Alg2):

Algorithm 1	Algorithm 2	Mean	Mode	Lower	Upper	P(Alg1 > Alg2)	P(Alg1 < Alg2)
EBGM_Base_FGPredictiveStep	ENERGY_PCA_MahCosine	-6.77	-7	-17	3	0.0724	0.8977

Table 11: Recognition statistics comparing the PCA and the EBGM algorithm.

The algorithms that perform better are shown in highlighted color, and the probabilities are only highlighted when the difference in recognition rate statistic is above a standard 95 percent confidence threshold. The paired test that is more discriminating, than, is simply checking for overlap in error bars. The primary reason for the difference is that the algorithms do better for a particular set of probe and gallery images that are harder for both the algorithms, while others are easier for both algorithms. Though the overall variation in recognition rates of the 10,000 trials is moderately large, the variation between the algorithms is somewhat smaller. More importantly the two algorithms are moving together based on the randomly generated probe and gallery sets. Hence, we observe that PCA is somewhat better than EBGM, but is not significantly better. However, we can get better recognition results with a hybrid technique that uses customized EBGM algorithm generating closest matching images within rank 50 instead of only rank 0. These best matching images will be rigorously matched further to obtain the best match.

5 SUMMARY

We studied the face recognition process and different techniques that are involved in achieving this task. The first step in the process is to acquire the image. The captured image is further processed to remove non-critical and biased data from the image. Preprocessing stage consist of different tasks such as Geometric normalization, masking, histogram equalization, pixel normalization, etc. After preprocessing the image the face recognition algorithm performs similarity measurement of the test image with the images in the gallery, the closest matching image is obtained as the best match. The face recognition algorithm builds and saves the distance measure for each gallery image. To generate the distance measure, the algorithm may either perform automated or semi-automated learning. Among the face recognition techniques that we studied, the PCA, the EBGM, and the Genetic Algorithm technique were the most outstanding techniques that were compared individually to each other with pros and cons. The face recognition technique using genetic algorithms is very interesting technique and can be explored further in separate paper.

The Principal Component Analysis (PCA) algorithm is one of the most prominent and efficient techniques to perform face recognition. Using a distance measure, we find the best matching image from the gallery for the given probe image. Each gallery image is compared with the probe image, and the gallery image that has lowest distance measure from the probe image is termed as the best matching image. Few of the commonly used distance measures are city-

block (L1), Euclidean (L2), and Mahalanobis cosine (MahCosine). Comparing the recognition results generated by the PCA technique with various distance measures, we found the best recognition using the Mahalanobis cosine distance. The CSU system provides a method to retain only critical eigenvectors that are: Simple, Energy, Stretch, and None. In the Simple technique, the PCA algorithm arranges the eigenvectors in ascending/descending order and then retains only certain percentage (default 40 percent) of the eigenvectors for further processing. In the Energy method, we retain only those eigenvectors that contribute to a particular percentage (default 90 percent) of the total eigenvector energy. While in the Stretch technique, the PCA algorithm only retains the eigenvectors that are greater than a percentage (default 1.00 percent) of the largest eigenvector. In contrast to the above methods, the None method retains all the eigenvectors. After obtaining critical eigenvectors, we can either retain all the eigenvectors (lossless) or eliminate (lossy) certain percentage of top/bottom eigenvectors. By eliminating these critical eigenvectors further we try to either eliminate noise or illumination. We obtained the best recognition results for the lossless Energy PCA technique in conjunction with the Mahalanobis cosine distance.

In the Elastic Bunch Graph Matching (EBGM) technique, the given image is represented as a graph. The nodes of this graph refer to the important points on the object's aspect and labeled by bunch of jets obtained from different face images. Edges of the graph are labeled with distance vectors between nodes. To form a bunch graph, a collection of facial images is marked with node locations

at defined positions of the head, landmarks. A semi-automatic process obtains these landmarks. When matching a bunch graph to an image, the jet extracted from the image is compared to all jets in the corresponding bunch attached to the bunch graph and the best matching one is selected. The EBGM algorithm is accomplished in three steps: face finding, landmark finding, and recognition by comparison. In face finding, the system tries to find a face in an image and determine its size that is accomplished by a set of matches to bunch graphs of appropriate pose and of three different sizes. *Landmark Finding* step finds facial landmarks with high positional accuracy and reliability, and encode the information contained in the image as accurately as possible. *Recognition by Comparison* does pair-wise comparison of the model graphs produced to the test image and obtain a similarity value. This similarity value is used to identify the best match between the test image and the gallery images. We studied the EBGM technique with the Wiskott, and Bolme wavelets. We found that the EBGM technique with Wiskott wavelet, predictive step learning and measurement gave best recognition results. The major drawback of the EBGM algorithm was the long time duration required to compute the distance matrix in comparison to the PCA algorithm. For the EBGM algorithm, addition of a new image and the process of obtaining the distance measures with reference to all other images in the gallery took less than three seconds. While, addition of a new image for the PCA algorithm require computing the distance measure that took just over nine seconds. As stated before, the PCA algorithm relies on Gaussian distribution, and addition of new images may generate biased data that

subsequently might lead to erroneous results. By re-computing the training vector we can solve this biased data problem. However, the time involved to find the training vector will hinder practical implementation of the PCA system. In contrast, the EBGM algorithm under similar situation took relatively less time.

6 APPENDIX

6.1 TIFFTOPGM Shell Script

```
#! /bin/sh
# Set this path to the current location where
# FERET images are kept in tif bz format
FERET_DIR='/home/sachin/thesis/feret/cd2/data/images/'
DEST_DIR='/feret/final_release/pgm/cdtwo/'
TMP_DIR='/tmp/convferet2pgm/'
LS='ls'
BUNZIP='/usr/bin/bunzip2'
MKDIR='mkdir'
CP='cp'
MV='mv'
RM='rm'
FILE='file'
GREP='grep'
PPMTOPGM='ppmtopgm'

#Since tifftopnm is the utility that converts the tif files to
#pgm we need this too

TIFF2PNM='tifftopnm'
echo "Starting conversion utility that converts FERET tif files to"
echo "Portable Grayscale Map (PGM) "
echo "Checking for source FERET image directory ...."

if [ -d $FERET_DIR ]
then
      echo "Directory $FERET_DIR exists. Continue..."
else
```

```
	echo "The specified directory ${FERET_DIR} does not exist. Please check
the path again."
	exit
fi

echo "Checking for destination FERET image directory ...."
if [ -d $DEST_DIR ]
then
	echo "Directory $DEST_DIR exists. Continue..."
else
	echo "The destination directory $DEST_DIR does not exist. Please check
the path again."
	exit
fi

if [ -d $TMP_DIR ]
then
	echo "Directory $TMP_DIR already exists"
else
	echo "Creating $TMP_DIR for processing purposes"
	echo `$MKDIR $TMP_DIR`
fi

for onefile in `$LS $FERET_DIR`
do
	echo "The file being processed is $onefile "
	#bunzip the file to some temp file name
	OUTPUT=`echo ${onefile} | sed 's/.bz2//g'`
	echo "Copying the ${FERET_DIR}${onefile} into
${TMP_DIR}${OUTPUT}"
	`${CP} ${FERET_DIR}${onefile} ${TMP_DIR}`
	 echo "bunzip 'ing the file "
	 echo "${BUNZIP} ${TMP_DIR}${OUTPUT}"
	`${BUNZIP} ${TMP_DIR}${onefile}`

	 #Run tifftopnm now
	PGMOUTPUT=`echo ${OUTPUT} | sed 's/.tif/.pgm/g'`

	 echo "Converting Tiff to PGM, saving as $PGMOUTPUT"
	echo "${TIFF2PNM} < ${TMP_DIR}${OUTPUT} >
${DEST_DIR}${PGMOUTPUT}"
	`${TIFF2PNM} < ${TMP_DIR}${OUTPUT} >
${DEST_DIR}${PGMOUTPUT}`

	echo "${FILE} ${DEST_DIR}${PGMOUTPUT}"
	FILE_TYPE=`${FILE} ${DEST_DIR}${PGMOUTPUT}`
```

```
        #echo $FILE_TYPE
        FILE_OUT=` echo "${FILE_TYPE}" | ${GREP} 'PGM'`
        #echo ${FILE_OUT}

        if [ "$FILE_OUT" = "" ]
        then
                #Since the file generated is not PGM it must be PPM so use
ppmtopgm
                PPMOUTPUT=`echo ${PGMOUTPUT} | sed 's/.pgm/.ppm/g'`
                # First move the PGM extension PPM file as PPM extension PPM file
                echo "$MV ${DEST_DIR}$PGMOUTPUT
${DEST_DIR}$PPMOUTPUT"
                `$MV ${DEST_DIR}$PGMOUTPUT ${DEST_DIR}$PPMOUTPUT`
                echo "$PPMTOPGM ${DEST_DIR}${PPMOUTPUT} >
${DEST_DIR}${PGMOUTPUT}"
                # USE PPMTOPGM command to convert ppm file to pgm format
                `$PPMTOPGM ${DEST_DIR}${PPMOUTPUT} >
${DEST_DIR}${PGMOUTPUT}`
                #Remove the ppm extension file
                echo "$RM -f ${DEST_DIR}${PPMOUTPUT}"
                `$RM -f ${DEST_DIR}${PPMOUTPUT}`
                #exit
        fi
        #DELETE temp file
        `$RM -f ${TMP_DIR}${OUTPUT}`
done
echo "All the FERET Files have been processed"
```

7 CONCLUSIONS AND FUTURE WORK

We have compared the lossless Energy cutoff PCA technique with Mahalanobis Cosine distance measure to the Wiskott wavelet based EBGM technique using predictive step learning and measurement. Probability distributions for recognition rates and differences in recognition rates relative to different choices of gallery and probe images have been created using a Monte Carlo sampling method. We observed that the PCA technique gave best result at rank 0, while as the rank increased the EBGM technique did far better. In spite

of the time and resource required for the EBGM technique, the EBGM technique does significantly well when the recognition rate is obtained over rank 50 instead of rank 0. As we learned from this paper, the PCA technique is sensitive to changes that are significantly affect the image such as occlusion, lightning, etc. While, the EBGM algorithm does not have such shortcomings. We can exploit this fact further for our use by obtaining the best matching results within rank 25 to rank 50 and then subjecting the obtained faces for exact match. In fact, we can use the results obtained with the EBGM algorithm and subject them to further processing to obtain precise match among the identified faces.

8 BIBLIOGRAPHY

[1] Bruce A. Draper, Kyungim Baek, Marian Stewart Barlett, J. Ross Beveridge, “Recognizing Faces with PCA and ICA”, Computer Vision and Image Understanding, Vol. 91, Issue 1/2, July 2003.

[2] Lixin Fan and Kah-Kay Sung, "Model-based varying pose face detection and facial feature registration in colour images", Pattern Recognition Letters, Volume 24, Issue 1-3, January 2003.

[3] W. Yambor, B. Draper, and R. Beveridge, "Analyzing PCA-based face recognition algorithms: Eigenvector selection and distance measures," 2nd Workshop on Empirical Evaluation in Computer Vision, 2000.

[4] Geof Givens, J. Ross Beveridge, Bruce A. Draper & David Bolme, "A Statistical Assessment of Subject Factors in the PCA Recognition of Human

Subjects", CVPR Workshop: Statistical Analysis in Computer Vision, June 22, 2003.

[5] K. Baek, B.A. Draper, J.R. Beveridge, K. She, "PCA vs. ICA: A comparison on the FERET data set", presented at Joint Conference on Information Sciences, Durham, NC, 2002.

[6] Kunio Takaya, Kyung-Yung Choi, "Detection of Facial Components In A Video Sequence By Independent Component Analysis", University of Saskatechewan, Canada.

[7] J.R. Beveridge, K. She, B.A. Draper, G.H. Givens, "A Nonparametric Statistical Comparison of Principal Component and Linear Discriminant Subspaces for Face Recognition", presented at IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, 2001.

[8] Ronald Alferez and Y. F. Wang, "Image Database Indexing using a Combination of Invariant Shape and Color Descriptions", The Second International Conference On Information Fusion, Sunnyvale, CA, July, 1999.

[9] Turaga, D.S. and Tsuhan Chen, "Face recognition using mixtures of principal components", Image Processing. 2002. Proceedings. 2002.

[10] A.J. Howell, H. Buxton, "Towards unconstrained face recognition from image sequences ", 2nd International Conference on Automatic Face and Gesture Recognition (FG '96), October 1996.

[11] Alper Yilmaz Mubarak A. Shah, "Automatic Feature Detection and Pose Recovery for Faces", proceedings of Asian Conference on Computer Vision, pp. 284-289, Australia, 2002.

[12] M. Kampmann, L. Zhang, "Estimation of eye, eyebrow and nose features in videophone sequences", International Workshop on Very Low Bitrate Video Coding (VLBV 98),pp. 101-104, Urbana, USA, October 1998.

[13] M.H.F. Wilkinson and J.B.T.M. Roerdink, "Fast morphological attribute operations using Tarjan's union-find algorithm ", In Mathematical Morphology and its Applications to Image and Signal Processing, J. Goutsias, L. Vincent and D. S. Bloomberg (eds.), Kluwer, pp. 311-320, 2000.

[14] Jun Zhang; Yong Yan; Lades, M., "Face recognition: eigenface, elastic matching, and neural nets ", Proceedings of the IEEE , Volume. 85, No. 9, pp. 1423 – 1435, Sept. 1997.

[15] Angel Kuri-Morales, "Holistic Face Recognition Through Multivariate Analysis and Genetic Algorithms ", Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning, Singapore, 2002.

[16] Xiaoling Wang, Hairong Qi, "Face Recognition Using Optimal Non-Orthogonal Wavelet Basis Evaluated by Information", 16th International Conference on Pattern Recognition (ICPR'02), Volume 1, Canada, August 2002.

[17] B. Fasel and J. Luettin, “Automatic facial expression analysis: a survey”, Tech. Rep. RR 99-19, IDIAP, Martigny, Valais, Switzerland, Dec. 2000. T.F. Cootes, G.J. Edwards, and C.J. Taylor, “Active appearance models,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681–685, June 2001.

[18] L. Torres J. Y. Reutter L. Lorente, "The Importance of the color information in face recognition ", University of Catalonia, Spain.

[19] S.Malassiotis, F.Tsalakanidou, N.Mavridis, V.Giagourta, N.Grammalidis, M.G.Strintzis, "A Face and Gesture Recognition System Based on an Active Stereo Sensor", IEEE International Conference on Image Processing (ICIP 2001), Thessaloniki, Greece, October 2001.

[20] Sally Wiener Grotta , "Anatomy of a Digital Camera: Image Sensors ", Extreme Tech, June 2001.

[21] Diego A. Socolinsky,Andrea Selinger, Joshua D. Neuheisel, "Face recognition with visible and thermal infrared imagery", Computer Vision and Image Understanding archive Volume 91 , No. 1-2, pp. 72 – 114, July 2003.

[22] Volker Blanz, Thomas Vetter, "Face Recognition Based on Fitting a 3D Morphable Model", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1063-1074, 2003.

[23] V. Blanz, S. Romdhani, T. Vetter, "Face Identification across Different Poses and Illuminations with a 3D Morphable Model", Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 2002.

[24] L. Torres L. Lorente Josep Vila, "Automatic face recognition of video sequence using Self-Eigenfaces", International conference on Image Processing, ICIP 99, Kobe, Japan, October 99.

[25] Michael Lyons, Andre Plante , Sebastien Jehan, Seiki Inoue , and Shigeru Akamatsu, "Avatar Creation using Automatic Face Recognition", Proceedings, ACM Multimedia 98, pp. 427-434, Bristol, England, Sept. 1998.

[26] Sing-Tze Bow, "Pattern Recognition and Image Processing", Second Edition, Marcel Dekker Inc, Newyork, Basel.

[27] Linda G. Shapiro, George C. Stockman, "Computer Vision", Prentice-Hall Inc, New Jersey, 2001, pp. 21-370.

[28] Ross Beveridge, David Bolme, Marcio Teixeira and Bruce Draper, “The CSU Face Identification Evaluation System User’s Guide: Version 5.0”, Computer Science Department Colorado State University, Colorado, May 2003

[29] Kazunori Okada, Johannes Steffens, Thomas Maurer, Hai Hong, Egor Elagin, Hartmut Neven, and Christoph von der Malsburg, “The Boschum/USC Face Recognition System and How it Fared in the FERET Phase III Test”, Computer Science Department and Center for Neural Engineering, University of Southern California, California.

[30] G.W. Cottrell and M. Fleming, “Face recognition using unsupervised feature extraction”.

[31] M. Lades, J. Vorbruggen, J.Buhmann, J.Lange, C.V.D Malburg, and R. Wurtz, “Distortion invariant object recognition in the dynamic link architecture”.

[32] T.Hastie and W. Stuetzle, “Principal curves”.

[33] N. Kambhatla and T. K. Leen, “Dimension reduction by local principal component analysis”.

[34] M.S. Barlett, H.M. Lades, and T.J.Sejnowski, “Independent component representation for face recognition,” presented at SPIE symposium on Electronic Imaging: Science and Technology; Conference on Human Vision and Electronic Imaging III, San Jose, CA, 1998.

[35] C. Liu and H. Wechler, “Comparative Assessment of Independent Component Analysis (ICA) for Face Recognition,” presented at International Conference on Audio and Video Based Biometric Person Authentication, Washington, D.C., 1999.

[36] B. Moghaddam, “Principal Manifolds and Bayesian Eigenfaces: Probabilistic Matching for Face Recognition,” presented at International Conference on Computer Vision, Corfu, Greece, 1999.

[37] Patrick Grother, FERET Technical Agent, NIST, Gaithersburg, MD, June 2001.

[38] NIST Website, http://www.itl.nist.gov/iad/humanid/feret/