Face Recognition Algorithms: Review, Benchmarking and Applications

Zhaoxian Zhou[1], *Hua Sun*[2], *and Chaoyang Zhang*[3]

Abstract – Face recognition has received substantial attention in recent years due to applications in research fields such as biometrics community and computer vision. A lot of face recognition algorithms have been developed during the past decades. These algorithms can be classified into appearance-based and model-based schemes. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) or Fisher Discriminant Analysis (FDA) are two typical linear appearance-algorithms, and Elastic Bunch Graph Matching (EBGM) is a two-dimensional model-based approach. This paper reviews the three classical methods and a typical face image database for standard testing. After the review is presented, the algorithms are implemented on Matlab environment. Scenarios and performance benchmarking are compared for each of the algorithms. The effectiveness and bottlenecks of each computation are discussed and possible improvements in different applications are given.

Keywords: PCA, LDA, EBGM, algorithm, implementation, benchmarking

INTRODUCTION

Face recognition has received substantial attention in recent years. A lot of face recognition algorithms have been developed during the past few decades. These algorithms can be classified into appearance-based and model-based schemes. The appearance-based algorithms can be further divided as linear and non-linear; the model-based algorithms can be further divided as 2D and 3D [1]. Principal Component Analysis (PCA) [2, 3] and Linear Discriminant Analysis (LDA) [4, 5], or Fisher Discriminant Analysis (FDA), are two typical linear appearance-based algorithms; Elastic Bunch Graph Matching (EBGM) [6] is a 2D model-based approach. PCA, LDA and EBGM are the main topics covered in this paper for the reason that they are the most widely used algorithms. They have attracted much more concern than other algorithms in the research areas such as machine learning and computer graphics. Graduate students in these fields should be familiar with the algorithms. Even for undergraduate students (seniors and juniors) in engineering, it is beneficial to understand the concepts in face recognition by implementing these algorithms following this paper.

Both PCA and LDA have different representations (i.e., basis vectors) of a high dimensional face vector space based on different statistical viewpoints. PCA algorithms use eigenfaces for dimensionality reduction to find the vectors that best account for the distribution of face images within the entire image space. The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible. LDA algorithms search for those vectors in the underlying space that best discriminate among classes. It seeks to find directions along which the classes are best separated. In both PCA and LDA, by projecting the face vector to the basis vectors, the projection coefficients are used as the feature representation of each face image. The distance between a test face image and the training prototype is calculated. If the distance is small enough, the image is recognized; otherwise, a new image is established. LDA takes into consideration the scatter within-classes and the scatter between-classes and thus is more capable of distinguishing image variation due to other sources such as illumination and expression. It is generally believed that LDA algorithms are superior to PCA algorithms, although it is not always the case. When the training data set is small, PCA may outperform LDA (this has been proved in our

[1] School of Computing, the University of Southern Mississippi, Zhaoxian.Zhou@usm.edu

[2] School of Computing, the University of Southern Mississippi, Hua.Sun@eagles.usm.edu

[3] School of Computing, the University of Southern Mississippi, Chaoyang.Zhang@usm.edu

experiment), and PCA is less sensitive to different training data sets. In EBGM algorithm, faces are represented as labeled graphs with nodes positioned at fiducial points (eyes, nose, mouth, and etc.) based on a Gabor wavelet transform. In a typical EBGM algorithm, a set of Gabor wavelet coefficients for each point is generated after the wavelet transform process. Several feature points representing the local features are extracted from the training faces. After that, for each feature point, a feature vector is generated by combining the Gabor wavelet transform coefficients with its coordinate. Every feature point is represented by a feature vector that includes a bunch of Gabor wavelet transform coefficients and its coordinate. Finally, all the feature vectors mentioned above are combined together to represent the face that is used in comparison and recognition process. EBGM is the best in terms of identification rate and performance reliability; however, poor illumination reduces recognition especially at nighttime.

After the introduction of the algorithms, the second section describes the implementation of PCA, LDA and EBGM. The next section explains the test method followed by the test results.

PCA

The goal of PCA is to find a best way to re-express the original data, and the re-expressed data can be described by a bunch of lower dimension basis vectors in a much more efficient way. The "best way" here means the noise and redundancy of the date should be as small as possible. In order to quantitatively express the data noise and redundancy, covariance matrix has to be introduced. In a case of an image with *m* pixels, the image can be expressed by an *m* row, one column matrix:

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \qquad (1)$$

Then, the covariance matrix is defined as:

$$C_X = \frac{1}{n-1} XX^T = \begin{bmatrix} \sigma^2_{x_1x_1} & \cdots & \sigma^2_{x_1x_m} \\ \vdots & \ddots & \vdots \\ \sigma^2_{x_mx_1} & \cdots & \sigma^2_{x_mx_m} \end{bmatrix} \qquad (2)$$

It is clear that the covariance matrix includes all the correlation information of the *m* pixels. The most important thing is that the correlation information reflects the noise and redundancy of data. The diagonal elements are the values of each pixel's variance. A relative larger value means that the element is important; a smaller value means that the element is less important or noise. The values of other elements describe the redundancy between each pixel. Therefore, the PCA problem can be expressed as finding a transform method that can make the value of the diagonal elements in the covariance matrix as large as possible and the other elements' value as small as possible. In the linear algebra language, this process is described as finding a transform matrix *p* that is composed by a bunch of orthogonal vectors to diagonalize the covariance of the transformed matrix. The transformed matrix is defined as:

$$Y=PX \qquad (3)$$

The covariance of Y is:

$$C_Y = \frac{1}{n-1} YY^T \qquad (4)$$

After certain linear algebra calculation [3], it is found that the orthogonal vectors or the basis vectors are the eigenvectors of XX^T. Therefore, by projecting the face image onto these basis vectors, the projection coefficients can be used as the feature representation of each image. The distance between a test face image and training prototype can be calculated. Then, the comparison and recognition process can be carried out based on the projection coefficients instead of original data vectors.

LDA

Both PCA and LDA have different representations (basis vectors) of a high dimensional face vector space based on different statistical viewpoints. PCA algorithm uses eigenfaces for dimensionality reduction to find the vectors that best account for the distribution of face image within the entire image space. LDA search for those vectors in the underlying space that best discriminate among classes (rather than those that best describe the data). More formally, given a number of independent features relative to which the data is described, LDA creates a linear combination of these that yields the largest mean differences between the desired classes.

To make the problem simple, we first consider two-class projection. The main purpose is to find a vector w to project the data onto this vector and get a new coordinate y, i.e.,

$$y = w^T x \qquad (5)$$

According to the spirit of LDA, we hope, by performing the projection, the data distance between the same classes is as small as possible and the distance between the different class is as large as possible. In order to describe in a quantitative way, some terms need to be defined. The mean value of each class data is

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x \qquad (6)$$

then, the mean value after the projection is

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} y = \frac{1}{n_i} \sum_{x \in D_i} w^T x = \frac{1}{n_i} w^T \sum_{x \in D_i} x = w^T m_i \qquad (7)$$

n_i is the total data number the i^{th} class, D_i is the set of the i^{th} class data, Y_i is the set of the i^{th} class data after the projection. We can define the average distance of the two data class after the projection as

$$|\tilde{m}_1 - \tilde{m}_2| = |w^T \, (m_1 - m_2) \, | \qquad (8)$$

We also define the scatter between each data class after the projection as

$$\tilde{s}_i^2 = \sum_{y \in Y_i} (\, y - \tilde{m}_i)^2 \qquad (9)$$

The objective of LDA is to find *w* that can maximize the value of the equation

$$J(w) = \frac{|\tilde{m}_1 - \tilde{m}_2|}{\tilde{s}_1^2 + \tilde{s}_2^2} \qquad (10)$$

Define the scatter matrix to describe the scatter of each class before projection

$$S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^T \qquad (11)$$

then every term of the denominator in equation (10) can be expressed as the combination of S_i and *w*:

$$\tilde{s}_i^2 = \sum_{x \in D_i} (\, w^T x - w^T m_i)^2 = \sum_{x \in D_i} w^T (x - m_i) \, (x - m_i)^T w = w^T S_i w \qquad (12)$$

The denominator in equation (10) can be expressed as

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T (S_1 + S_2) w = w^T S_W w \qquad (13)$$

and the numerator can be expressed as

$$(\tilde{m}_1 - \tilde{m}_2)^2 = (w^T m_1 - w^T m_2)^2 = w^T (m_1 - m_2)(m_1 - m_2)^T w = w^T S_B w. \qquad (14)$$

Therefore, $J(w)$ is rewritten as

$$J(w) = \frac{w^T S_B w}{w^T S_W w}, \qquad (15)$$

where S_B and S_W are named as between-matrix and within-matrix.

If the class number *c* is larger than 2, the between-matrix is then expressed as

$$S_B = \sum_{i=1}^{c} n_i \, (m_i - m)(m_i - m)^T, m = \frac{1}{n} \sum_x x, \qquad (16)$$

and the within-matrix is expressed as

$$S_W = \sum_{i=1}^{c} S_i \qquad (17)$$

The LDA problem can be described as finding a set of *M* feature basis vectors, denoted as $\{\Psi_m\}_{m=1}^{M}$, in such a way that the ratio of the between- and within-class scatters of the training sample is maximized. The maximization problem is generally formulated as:

$$\Psi = \arg\max_{\Psi} \frac{|\Psi^T S_B \Psi|}{|\Psi^T S_W \Psi|}, \Psi = [\psi_1, \cdots, \psi_M] \tag{18}$$

The optimization problem of the equation above is equivalent to the following generalized eigenvalue problem,

$$S_B \psi_m = \lambda_m S_W \psi_m, m = 1, \cdots, M \tag{19}$$

Thus, the basis vectors are corresponded to the first M most significant eigenvectors of $({S_W}^{-1} S_B)$.

EBGM

EBGM algorithm directly uses the local features of human faces to solve face recognition problems. However, not like the features we can always capture by using our eyes, such as nose, eyes, and mouth. The features used in EBGM algorithm are the wavelet transforms of the original data that represent the notable features of faces. In other words, both human and computer can distinguish one face from the others by capturing these notable features; however, the difference is the tools that have been used: human use their eyes, computer uses wavelet transforms. Compared to PCA and LDA, the re-expression of original image by using EBGM is relatively easy to understand. Some feature points are extracted after performing the Gabor wavelet transform, and these feature points, instead of original image, are used in the comparison and recognition process. We can see from Figure 1 that the feature points are concerning the eye, mouth and face shape and these features are substantial in the case distinguishing one people from another.

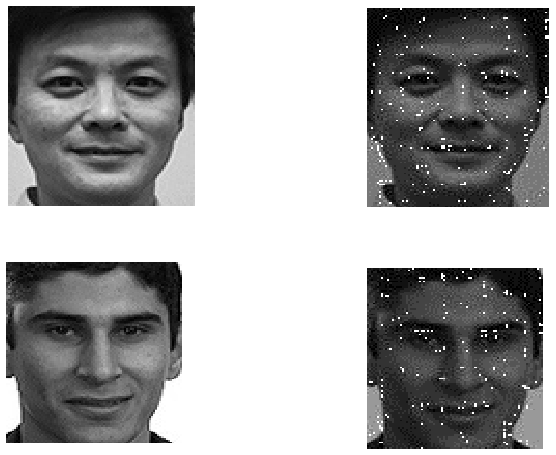

Figure 1: Example of face feature points extraction

Algorithm Implementation

PCA and LDA are very similar algorithms from the aspect that they both try to find lower dimensional basic vectors to represent the original data in the form of linear combination of these basis vectors. The difference is that PCA is used to find a subspace whose basic vectors correspond to the maximum-variance directions in the original space; LDA, however, searches for those vectors in the underlying space that best discriminate among classes rather than those that best describe the data. The subspaces searched by PCA and LDA are named as eigenfaces and fisherfaces respectively.

PCA implementation

PCA finds a set of most representative projection vectors to retain most information about the original image. Based on eigenface algorithms [2, 3], the Matlab implementation of PCA is described as follows.

Input: A training set Γ, which includes a number of images for each subject, with some variations in expression and lighting (say two images of fifty subjects, M=100)

Output: A PCA subspace spanned by μ.

Algorithm:

1. Calculate average face $\mathbf{T} = \frac{1}{M}\sum_{n=1}^{M} \mathbf{\Gamma}_n$.
2. Calculate the difference of each face from the average face $\mathbf{\Phi}_i = \mathbf{\Gamma}_i - \mathbf{T}$.
3. Define matrix A= $[\mathbf{\Phi}_1\ \mathbf{\Phi}_2\ \mathbf{\Phi}_3\ \cdots \mathbf{\Phi}_M]$.
4. Calculate matrix D=$A^T A$.
5. Find out the eigenvectors and eigenvalues of D as **v** and λ.
6. Calculate the eigenfaces by $\mu = A\mathbf{v}^T$.
7. Choose the M' eigenvectors with the highest associated eigenvalues as μ by setting up a threshold, return the principle eigenfaces μ.

LDA implementation

LDA uses class information to find a set of vectors to maximize between-class but minimize within-class scatter. Based on fisherface algorithm [4, 5], the Matlab implementation of LDA is described as follows.

Input: A training set Γ with C classes: $\Gamma=\{\Gamma_i\}_{i=1}^{C}$, each class containing $\Gamma_i = \{\Gamma_{ij}\}_{j=1}^{C_i}$ face images, where $\Gamma_{ij} \in \mathbb{R}^J$, and η is the regularization parameter.

Output: An M-dimensional LDA subspace spanned by Ψ, a $J\times M$ matrix with $M \ll J$.

Algorithm:

1. Express $\mathbf{S}_b = \Phi_b\Phi_b^{\mathrm{T}}$, with $\Phi_b = [\Phi_{b,1}, \cdots, \Phi_{b,c}]$, $\Phi_{b,i} = (C_i/N)^{1/2}(\overline{\Gamma}_i - \overline{\Gamma})$, $\overline{\Gamma}_i = 1/C_i \sum_{j=1}^{C_i} \Gamma_{ij}$, and

   $\overline{\Gamma} = 1/N \sum_{i=1}^{C}\sum_{j=1}^{C_i} \Gamma_{ij}$.
2. Find the m eigenvectors of $(\Phi_b^T\Phi_b)$, with non-zero eigenvalues, and denote them as $\mathbf{E}_m = [\mathbf{e}_1, \cdots, \mathbf{e}_m]$.
3. Calculate the first m most significant eigenvectors $(\mathbf{U}_m)$ of $\mathbf{S}_b$ and their corresponding eigenvalues (Λ_b) by

   $\mathbf{U}_m = \Phi_b\mathbf{E}_m$ and $\Lambda_b = \mathbf{U}_m^T\mathbf{S}_b\mathbf{U}_m$.
4. Let $\mathbf{H}$=$\mathbf{U}_m\Lambda_b^{-1/2}$. Find eigenvectors of $\mathbf{H}^T\mathbf{S}_w\mathbf{H}$, $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_m]$ sorted in increasing eigenvalues order.
5. Choose the first M $(\leq m)$ eigenvectors in **P**. Let $\mathbf{P}_M$ and Λ_w be the chosen eigenvectors and their corresponding eigenvalues, respectively.
6. Return $\mathbf{\Psi} = \mathbf{HP}_M(\eta\mathbf{I} + \Lambda_w)^{-1/2}$.

EBGM implementation

In EBGM algorithm, faces are represented as labeled graphs, with nodes positioned at fiducial points (eyes, nose, mouth, and *etc*.) based on a Gabor wavelet transform. EBGM is the best in terms of identification rate and performance reliability, however, poor illumination reduces recognition especially at nighttime. The EBGM algorithm used here is based on [8]. By using this algorithm, a set of Gabor wavelet coefficients for each point is generated after the wavelet transform process. Several feature points representing the local features are extracted from the training faces. After that, for each feature point, a feature vector is generated by combining the Gabor wavelet transform coefficients with their coordinate. Every feature point is represented by a feature vector that includes a bunch of Gabor wavelet transform coefficients and their coordinate. Finally, all the feature vectors mentioned above are combined together to represent face that is used in comparison and recognition process.

Comparing to other algorithms, such as [9, 10], the algorithm used here has two advantages: there is no need for manual localization of the training graphs and it is easy to implement. One drawback is that, to significantly reduce the computation cost, the face images need to be well segmented in order to remove the redundant information concerning with hair and background from the original images. The implementation of EBGM is described as follows.

Input: A training image I, which is described as I =$[I(\mathbf{x}_1), I(\mathbf{x}_2), \cdots, I(\mathbf{x}_\mathrm{n})]$, where $I(\mathbf{x}_\mathrm{i})$ is the image intensity value at $\mathbf{x}_\mathrm{i}$, and $\mathbf{x}_\mathrm{i}$ is the coordinate vector of point I, n is the total point number

Output: A bunch of feature vectors described as $v_k = \{x_k, y_{k,} R_j(x_k, y_{k,})\ j = 1, \cdots 40\}$, which represents the k^{th} feature vector of the reference face image. $R_j(x_k, y_{k,})$ is the Gabor wavelet transform coefficients at the $(x_k, y_{k,})$ point.

Algorithm:

1. Applying Gabor wavelet transform to image I, $R_i(\mathbf{x}) = \int I(\mathbf{x}')\Psi_i(\mathbf{x} - \mathbf{x}')d^2\,\mathbf{x}'$, $I(\mathbf{x})$ is the intensity value at $\mathbf{x}$, and the Gabor filter is expressed as:

$$\Psi_i(\mathbf{x}) = \frac{\|\mathbf{k}_i\|^2}{\sigma^2} e^{-\frac{\|\mathbf{k}_i\|^2\|\mathbf{x}\|^2}{2\sigma^2}} \left[e^{j\,\mathbf{k}_i \cdot \mathbf{x}} - e^{-\frac{\sigma^2}{2}}\right] \tag{20}$$

where

$$\mathbf{k}_i = \begin{pmatrix} k_{ix} \\ k_{iy} \end{pmatrix} = \begin{pmatrix} k_v \cos\theta_\mu \\ k_v \sin\theta_\mu \end{pmatrix},\ k_v = 2^{-\frac{V+2}{2}}\pi,\ \theta_\mu = \mu\frac{\pi}{8},\ v{=}0,\cdots,4,\ \mu = 0,\cdots,7,\ i = \mu + 8v \tag{21}$$

2. Find out feature points by searching the location in a window W_0 of size $w \times\ w$ by the following procedure:

A feature point is located at (x_0, y_0) if

$$R_j(x_0, y_0) = max\left(R_j(x, y)\right) \tag{22}$$

and

$$R_j(x_0, y_0) > \frac{1}{N_1 N_2}\sum_{x=1}^{N_1}\sum_{y=1}^{N_2} R_j(x, y) \tag{23}$$

where $(x, y) \in W_0$, j=0,$\cdots$,40 and $N_1 N_2$ is the size of the face image. The window size should be chosen small enough to capture the important features and large enough to avoid redundancy.

3. Generate feature vectors at feature points as a composition of Gabor wavelet transform coefficients:

$$v_k = \{x_k, y_{k,} R_j(x_k, y_{k,})\ j = 1, \cdots 40\} \tag{24}$$

PERFORMANCE EVALUATION

In order to compare the performance of different algorithms, images databases are needed. Some notable databases include (1.) The Color FERET Database, USA (14,126 images of 1199 individuals); (2.) The Yale Face Database (165 grayscale images in gif format of 15 individuals). Others include databases from AT&T, University of Essex, and Caltech. The database employed in this paper is a fraction of the FERET database. We use 100 male and female faces of 50 individuals, i.e., each subject has 2 images (represented as "A" and "B") with different expressions, illumination, and /or accessories.

Test Process

The Matlab codes run on a HP desktop computer with Pentium Dual-Core CPU at 2.6 GHz, 4 GB of memory, and 64-bit Windows 7 Home Premium operating system.

The 100 images are divided into two groups, the training set containing the 50 "A" images and the testing set containing the 50 "B" images. After the implementation of these three algorithms, some feature parameters are extracted to represent the original image and will be used in the comparison and recognition phase. For PCA and LDA, these feature parameters are sets of coefficients generated by projecting the image data onto the basis vectors; for EBGM, these feature parameters are some feature points of the face expressed by some Gabor wavelet transform coefficients. In the comparison phase, the feature parameters of each image from each group are used to calculate the distance or similarity. To be more specific, the distance when using PCA and LDA, or the similarity when using EBGM of each test image with other 50 training images are calculated. In an ideal situation, the distance between two images representing the same subject should be much smaller than those representing different subjects, as seen in Figure 2, or the similarity, inversely, should be much larger than others, seen as in Figure 3.

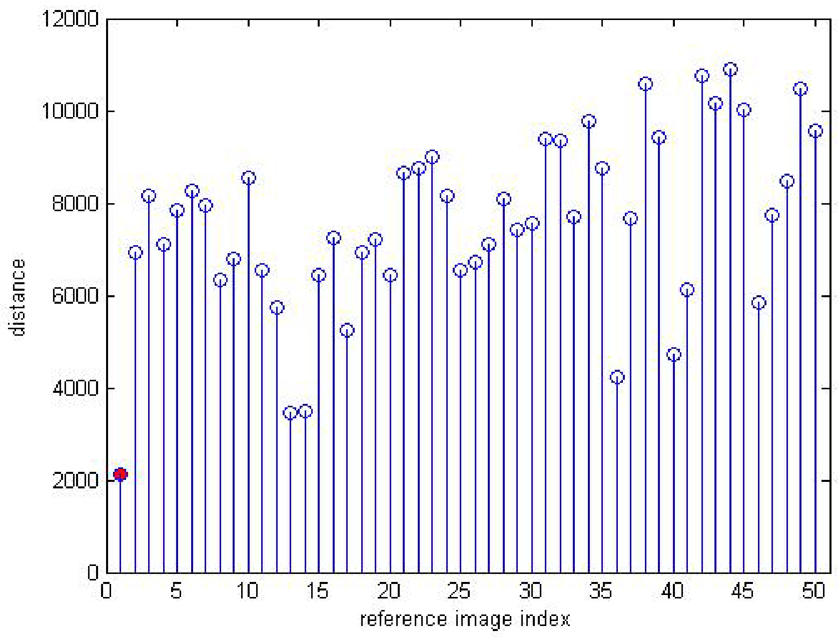

Figure 2: Distances between probe image and reference images

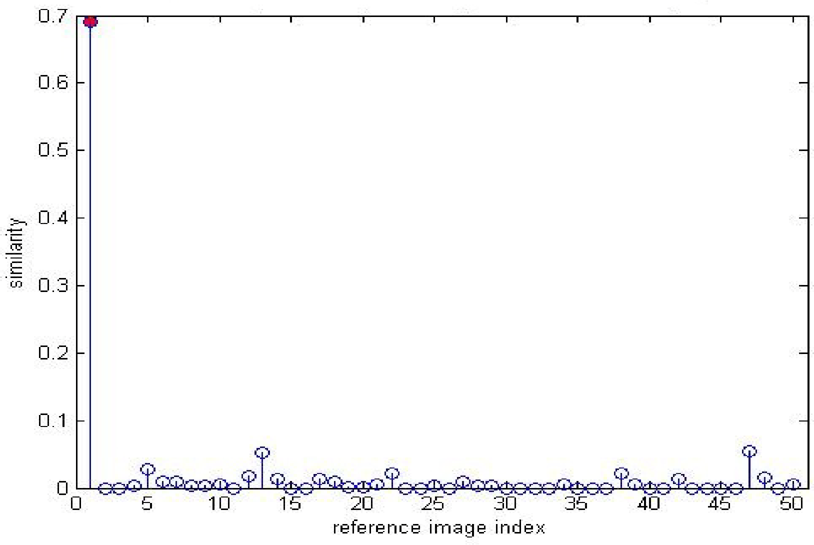

Figure 3 Similarity between probe image and reference images

Test Result

We will analyze performance of the algorithms from three points of view: recognition accuracy rate, computational cost, and perturbation tolerance.

1. Recognition accuracy rate: this is one of the most important parameter used to evaluate the performance of face recognition algorithm. Following the test process stated above, the accuracy can be easy calculated. For PCA and LDA, the accuracy rates are derived from the comparison of distance between different images. If two images have the smallest distance, and also represent the same people in different expression,

then we say this is successful recognition. After the comparison phase, the whole accuracy can be calculated as the ratio of total successful comparison number and the whole comparison number. The situation for EBGM is very similar; the only difference is that we use similarity instead of distance.

2. Computational time: this is crucial for any real-time recognition system. The test result can be seen in table 1. We can see from table 1 that PCA runs fast but the accuracy rate is relative low. EBGM has the highest accuracy rate; however, the processing time is much longer than those of the other two algorithms. LDA has both good accuracy rate and small computational time.

Algorithm	Training Images	Accuracy Rate (%)	Processing Time (s)
PCA	50	68	38
LDA	50	74	40
	100 (two class each)	91	73
EBGM	50	97	3540

Table 1: Test result

3. Recognition tolerance. If an algorithm distinguishes an image correctly, the distance should be the smallest or the similarity should be the largest among other distances or similarities. However, how small is the distance or how large is the similarity? This figure represents the ability of an algorithm to recognize the correct subject from a set of similar images. Figure 4 plots the similarity between one testing image and all training images. We see that the maximum similarity is prominent compared to other values. In Figure 5, the distance between the same image and all training images are plotted. We see that although the correctly-recognized image indeed has the smallest distance, the distance value of recognized images is only slightly smaller than others. This means the image cannot distinguish itself from others very well; the recognition process may fail if the expression difference or physical difference of the two images was a little larger. In other words, EBGM has a better recognition tolerance.

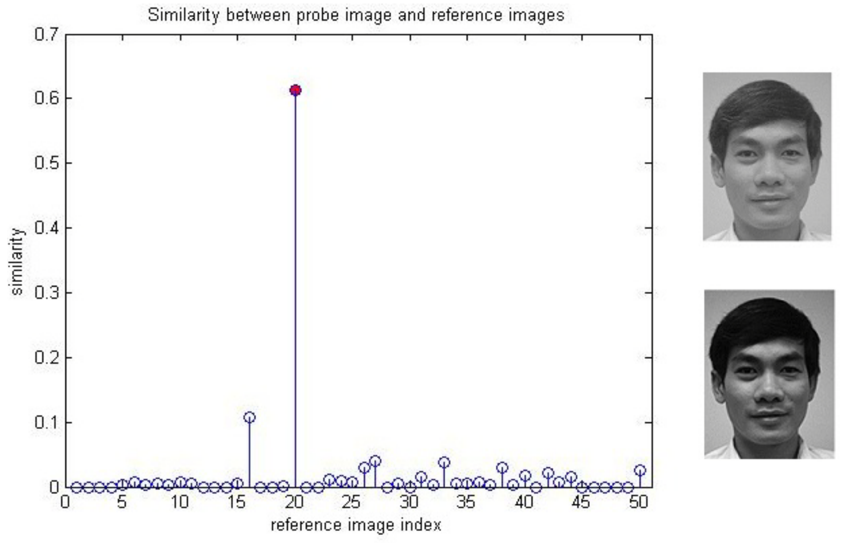

Figure 4: Face recognition by EBGM

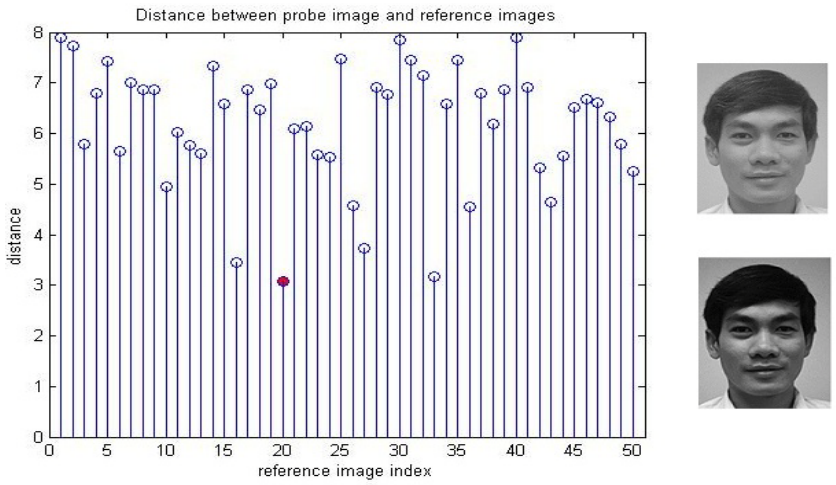

Figure 5: Face recognition by LDA

Discussions

Basically, PCA and LDA are two principle algorithms used for decreasing data dimension. The difference is PCA processes the data in a way to minimize the noise and redundancy, and LDA aims to maximize the distance between different classes. That is why PCA is widely used in data compression and LDA gets a good reputation in pattern classification. When specifically applied to face recognition, LDA seems to perform better than PCA, no matter concerning recognition accuracy or processing speed, the only advance for PCA may be its simpler algorithm realization. Actually, the statement that LDA is always much more outstanding than PCA is not true. The prerequisite for a good performance of LDA is that there is a large number of samples for each class, which is cannot always be ensured in practice. In other words, when only a small amount of samples provided for each class, the recognition accuracy of LDA will decrease dramatically, sometimes is even worse than that of PCA. A possible improvement is combining these two algorithms together, applying PCA as a pre-process of the images for LDA. Doing this the whole recognition accuracy is advanced without significant increase in process time.

For EBGM, a crucial problem is computation complexity, which is related to the Gabor Wavelet transform. So, good face segmentation is necessary when applying EBGM for the reason the meaningless data, such as image background and hair will be removed. Another factor that affects the performance of EBGM is image size. Test result has shown that a quite large image size will worsen the recognition accuracy and, with no doubt, increase he process time. So far, to get a best result, the image size of a regular individual image as shown in figure 5 used in our lab is 50 times 50 pixels which is much smaller than that of original images.

Conclusion And Future Works

After implementation and comparison of three classical face recognition algorithms, PCA, LDA and EBGM, a brief performance result is reported in this paper. Three main factors, total training image number, accuracy rate and processing speed are considered when evaluating the performance. We can see from the result that both LDA and EBGM perform better than PCA. EBGM has the highest accuracy rate but the computational time is relative long. LDA performs with both high accuracy rate and small computational time; however, the recognition tolerance is weaker than EBGM. Our future work will focus on finding out other factors that may affect the performance of these three algorithms, such as image size and normalization methods.

References

[1] Xiaoguang Lu, "Image analysis for face recognition a brief survey," Personal Notes, May 2003.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience* 3, pp. 71–86, 1991.

[3] M. Turk, and A. Pentland, “Face recognition using eigenfaces,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (Maui, HI), June 3-6 1991.

[4] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Face recognition using LDA-based algorithms,” *IEEE Trans. Neural Networks* 14, pp. 195–200, January 2003.

[5] J. Lu, K. Plataniotis, and A. Venetsanopoulos, “Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition,” *Pattern Recognition Letter* 26.

[6] L. Wiskott, J.-M. Fellous, N. Kruger, and C. von der Malsburg, “Face recognition by elastic bunch graph matching,” *IEEE Trans. Pattern Analysis and Machine Intelligence* 19, pp. 775–779, July 1997.

[7] A. M. Martoznez and A. C. Kak, “PCA versus LDA,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23, pp. 228–233, Febrary 2001.

[8] B. Kepenekci, “Face recognition using gabor wavelet transform,” Master’s thesis, The Middle East Technical University, Sep., 2001.

[9] P. Phillips, “The feret database and evaluation procedure for face recognition algorithms,” *Image and Vision Computing* 16, pp. 295–306, 1998.

[10] L. Wiskott, J.-M. Fellous, N. Kruger, and C. von der Malsburg, *Face Recognition by Elastic Bunch Graph Matching*, pp. 355–396. CRC Press, 1999.

Zhaoxian Zhou
Dr. Zhou received the B. Eng. from the University of Science and Technology of China in 1991; M. Eng. from the National University of Singapore in 1999 and the PhD degree from the University of New Mexico in 2005. All degrees are in Electrical Engineering. From 1991 to 1997, he was an Electrical Engineer in China Research Institute of Radiowave Propagation. In the fall of 2005, he joined the School of Computing, the University of Southern Mississippi as an assistant professor. His research interests include electromagnetics, radiowave propagation, high performance computing and numerical analysis. His teaching interests include communications, electromagnetics, antennas and propagation, electric power, and signal processing. He is a senior member of IEEE and a member of ASEE.

Hua Sun
Mr. Sun received the BS in Communication Engineering from Harbin Institute of Technology in 2006, and currently is pursuing his Master’s degree in engineering technology in the School of Computing at the University of Southern Mississippi. He served as a RF Engineer from 2006 to 2008 in HeBei YuanDong Communication System Engineering Corporation. He Designed, tested and tuned over 20 RF modules in nearly 30 projects independently, including frequency synthesizers, filters, power combiners and splitters, frequency multipliers. His present research work deals with face recognition algorithm design and testing.

Chaoyang Zhang
Dr. Chaoyang Zhang received his PhD at Louisiana Tech University in 2001 and was a research assistant professor in the Department of Computer Science at the University of Vermont from 2001 to 2003. Currently he is the director and associate professor in the School of Computing, University of Southern Mississippi. His research interests include 3D image reconstruction, high performance computing, and data mining and computational biology. His research is supported by NSF, NIH and DOD and he has more than 40 peer-reviewed publications. He is the Program Committee chair of the 2009 IJCBS and Steering Committee co-chair of the ACM-BCB conferences.