Volume 2, No. 7, July 2011
Journal of Global Research in Computer Science
TECHNICAL NOTE
Available Online at www.jgrcs.info
ISSN-2229-371X

# COMPARISON BETWEEN FACE RECOGNITION ALGORITHM-EIGENFACES, FISHERFACES AND ELASTIC BUNCH GRAPH MATCHING

Sushma Jaiswal*[1], Dr. (Smt.) Sarita Singh Bhadauria[2], Dr. Rakesh Singh Jadon[3]

[1]Lecturer, S.O.S. in Computer Science, Pt. Ravishankar Shukla University,Raipur(C.G.)
jaiswal1302@gmail.com
[2]Prof. & Head, Department of Electronics, M. I. T. S. Gwalior, M. P. – 474005, India
[3]Prof. & Head, Department of Computer Application, M. I. T. S. Gwalior, M. P. – 474005, India

***Abstract:*** The technology of face recognition has become mature within these few years. System, using the face recognition, has become true in real life. In this paper, we will have a comparative study of three most recently methods for face recognition. One of the approach is eigenface, fisherfaces and other one is the elastic bunch graph matching. After the implementation of the above three methods, we learn the advantages and Disadvantages of each approach and the difficulties for the implementation.

***Keywords -*** face recognition**,** Eigenfaces, Fisherfaces and Elastic Bunch Graph Matching, advantages, Disadvantages.

.

## INTRODUCTION

The study of biometrics is becoming important in last 35 years. The study of biometrics is becoming important in last 35 years In the modern information age, human's information become valuable. It can be used for the security and important social issues. Therefore, identification and authentication methods have developed into a main technology in various areas, such as entrance control in building and access control for computers.

Most of these methods have a drawback with their legitimate applications. Except for the human and voice recognition, these methods almost require user to remember a password, or human action in the course of identification or authentication. However, the corresponding means are potential being lost or forgotten, whereas fingerprints and retina scans suffer from low user acceptance rate.

Face recognition has a high identification or recognition rate of greater than 90% for huge face databases with well-controlled pose and illumination conditions. This high rate can be used for replacement of lower security requirement environment and could be successfully employed in different kind of issues such as multi-media.

Automatic recognition is a vast and modern research area of computer vision, reaching from face detection, face localization, face tracking, extraction of face orientation and facial features and facial expressions. These will need to tackle some technical problems like illumination, poses and occlusions.

In this paper, we will focus on two recently used techniques in face recognition. The three techniques are eigenface by Alex P. Pentland [1] ,fisherface method of face recognition as described by Belhumeur et al [4] and elastic bunch graph matching by Laurenz Wiskott [3]. These techniques are recent and have apparently promising performances, and are representative of new trends in face recognition.

An automated face recognition system needs to overcome several difficulties.lighting condition is another major problem for face recognition. The same person under different lighting condition may be seen quite different. the same person seen under different lighting conditions can appear dramatically different. We almost cannot recognize two people even with our eyes. Facial expression will also make a face varies. All the problems mentioned above will dramatically decrease the accuracy of a face recognition system. For a reliable face recognition system, it should be accurate, efficient and invariant to changes. Accuracy is an important measurement of a face recognition system. For an accurate face recognition system, the accuracy should be over 79%. Otherwise, we cannot correctly recognize a person. Efficiency is critical for a real-time face recognition system. Users cannot tolerate a slow system to recognize a person or wait for the result of searching. The storage should also not be too large. It is not practical to store huge amount of data.

Besides, a face recognition system should overcome the rotational, intensity changes mentioned before. The system should work properly even the person has little head rotation or under moderate variation in lighting direction, brightness. Otherwise, the system can only be used under some specify conditions which makes it inflexible.

## ALGORITHMS

Within last several years, there are numerous face recognition algorithms written by researchers. Different approach likes neural networks, face unit radial basis function networks are proposed. In the following part of this paper, we would describe three algorithms that make use of feature extraction. The first two algorithms, Eigenface and Fisherface and third algorithm Elastic Bunch Graph Matching .

### *Eigen face*

Eigenface was suggested [1]. The main idea of eigenface is to get the features in mathematical sense instead of physical face feature by using mathematical transform for recognition.

There are two phases for face recognition using eigenfaces. The first phase is the training phase. In this phase, a large group of individual faces is acted as the training set. These training images should be a good representation of all the faces that one might encounter. The size, orientation and light intensity should be standardized. For example, all images are of size 125 x 125 pixels and all are frontal faces. Each of the images in the training set is represented by a vector of size *N* by *N*, with *N* representing the size of the image. With the training images, a set of eigen-vectors is found by using Principal Component Analysis (PCA).

The basic idea of PCA is to take advantages of the redundancy existing in the training set for representing the set in a more compact way. Using PCA, we can represent an image using *M* eigenvectors where *M* is the number of eigenvector used. Let the training set of face images be $\Gamma_1$, $\Gamma_2$, $\Gamma_3$, …, $\Gamma_M$. The average face of the set if defined by $\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$. Each face differs from the average by the vector $\Phi_n = \Gamma_n - \Psi$. An example training set is shown in Figure 1a, with the average face $\Psi$ shown in Figure 1b. This set of very large vectors is then subject to principal component analysis, which seeks a set of *M* orthonormal vectors, $\mu_n$, which best describes the distribution of the data. The *k*th vector, $\mu_k$ is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(\mu_k^T\Phi_n)^2$$

is a maximum, subject to

$$\mu_l^T\mu_k = \begin{cases} 1, l = k \\ 0, otherwise \end{cases}$$

The vectors $\mu_k$ and scalars $\lambda_k$ are the eigenvectors and eigenvalues, respectively, of the covariance matrix

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T$$

where the matrix $A = [\Phi_1\Phi_2 \ldots \Phi_M]$. The matrix *C*, however, is $N^2$ by $N^2$, and determining the $N^2$ eigenvectors and eigenvalues is an intractable task for typical image sizes. A computationally feasible method is needed to find these eigenvectors.

If the number of data points in the image space is less than the dimension of the space ($M < N^2$), there will be only $M - 1$, rather than $N^2$, meaningful eigenvectors (the remaining eigenvectors will have associated eigenvalues of zero). Fortunately, we can solve for the $N^2$-dimensional eigenvectors in this case by first solving for the eigenvectors of and *M* by *M* matrix—e.g., solving a 15 x 15 matrix and then taking appropriate linear combinations of the face images $\Phi_n$. Consider the eigenvectors $\nu_n$ of $A^TA$ such that

$$A^TA\nu_n = \lambda_n\nu_n$$

Premultiplying both sides by *A*, we have

$$AA^TA\nu_n = \lambda_nA\nu_n$$

from which we see that $A\nu_n$ are the eigenvectors of $C = AA^T$.

Following this analysis, we construct the *M* by *M* matrix $L = A^TA$, where $L_{mn} = \Phi_m^T\Phi_n$, and find the *M* eigenvectors $\nu_n$ of *L*. These vectors determine linear combinations of the *M* training set face images to form the eigenfaces $\mu_n$:

$$\mu_n = \sum_{k=1}^{M}\nu_{nk}\Phi_k = A\nu_n, n = 1,\ldots\ldots, M$$

With this analysis the calculations are greatly reduced, from the order of the number of pixels in the images ($N^2$) to the order of the number of images in the training set (*M*). In practice, the training set of face images will be relatively small ($M < N^2$), and the calculations become quite manageable. The associated eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizeing the variation among the images.

***Fisher face***

The fisherface method of face recognition as described by Belhumeur et al [4] uses both principal component analysis and linear discriminant analysis to produce a subspace projection matrix, similar to that used in the eigenface method. However, the fisherface method is able to take advantage of ëwithin-classí information, minimising variation within each class, yet still maximising class separation. Like the eigenface construction process, the first step is take each (NxM) image array and reshape into a ((N*M)x1) vector.

Next, using the $x_k$ values, calculate both the class mean $\mu_k$ and the mean of all the samples $\mu$

$$\mu = \frac{1}{N}\sum_{k=1}^{N}x_k = Mean$$

$$\mu_k = \frac{1}{N_k}\sum_{m=1}^{N_k}x_{k_m}$$

Where:

***N*** = Total Number of images

$N_k$ = Number of images in class k

$x_{k_m}$ = Image at index m of class k

Now we determine both the between-class scatter matrix ($S_B$) and the within-class scatter matrix ($S_W$).

$$S_B = \sum_{k=1}^{C}N_k(\mu_k - \mu)(\mu_k - \mu)^T$$

$$S_W = \sum_{k=1}^{C} \sum_{x \in X_k} (x_k - \mu_k)(x_k - \mu_k)^T$$

Where :

***C*** = Number of classes

The optimal eigenvectors ($U_{opt}$) can be found by the equation:

$$U_{opt} = \arg\max \frac{\left|U^T S_B U\right|}{\left|U^T S_W U\right|} = \left[u_1, u_2, \ldots, u_m\right]$$

This equation can then be simplified into a generalized eigenvalue equation:

$S_B u_i = \lambda_i S_W u_i : i = 1,2,\ldots, m$ Feature vectors can then be established using the equation

$$y_k = U^T x_k : k = 1,2\ldots, m$$

***Fisherface Concept***-Differing from the Eigenface concept, the fisherface method tries to maximize the ratio of the between-class scatter versus the within-class scatter [2]. The result of this shapes the projections so that the distances between the classes are at a maximum, while the distances between samples of the same class are at a minimum. A possible disadvantage is if the between-class scatter is large, then the within-class scatter might also still be of a relatively large value.

***Elastic Bunch Graph Matching***

Face recognition using elastic bunch graph matching [3] is based on recognizing novel faces by estimating a set of novel features using a data structure called a bunch graph. Similarly for each query image, the landmarks are estimated and located using bunch graph. Then the features are extracted by convolution with the number of instances of Gabor filters followed by the creation of face graph. The matching score (*MS<sub>EBGM</sub>*) is calculated on the basis of similarity between face graphs of database and query image. Elastic Bunch Graph Matching was suggested by Laurenz Wiskott, Jean-Marc Fellous, Norbert Kruger and Christoph von der Malsburg of University of Southern California in 1999. This approach takes into account the human facial features and is totally different to Eigenface and Fisherface. It uses elastic bunch graph to automatically locate the fiducial points on the face (eyes, nose, mouth etc) and recognize the face according to these face features.

Elastic Bunch Graph Matching (EBGM) uses the structure information of a face which reflects the fact that the images of the same subject tend to translate, scale, rotate, and deform in the image plane. It makes use of the labeled graph, edges are labeled the distance information and nods are labeled with wavelet coefficients in jets. This model graph can then be used to generate image graph. The model graph can be translated, scaled , rotated and deformed during the matching process. Gabor.

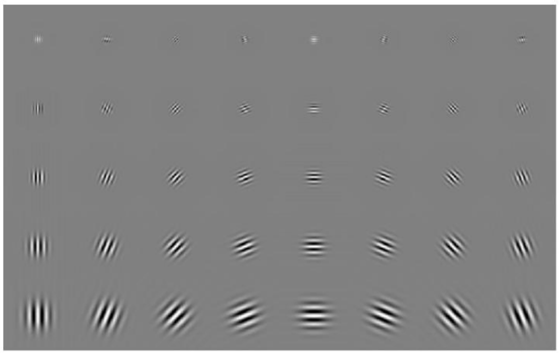

Figure 1 - The real part of the Gabor filter with 5 frequencies and 8 orientations

Gabor wavelet transformation is used to represent the local features of the face images. Gabor wavelets are biologically motivated convolution kernels in the shape of plan waves restricted by a Gaussian envelop function, the set of convolution coefficients for kernels of different orientations and frequencies at one image pixel is called a jet.

***Face Bunch Graph***-Automatic finding fiducial points in new faces need a general representation rather than models of the individual faces. A wide range of possible variations in the appearance of faces, like different shaped eyes, mouth, variation due to sex, age, etc, should be covered.

Combination each feature by a separate graph is not efficient. So we use a stack like structure called a face bunch graph **(FBG)**

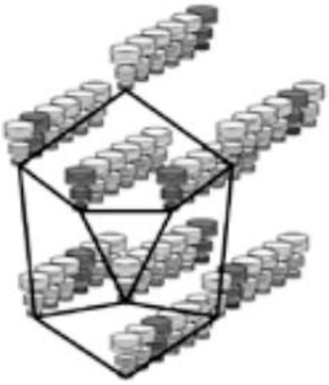

Figure 2– The face Bunch Graph represent the face in general.

The representation of facial feature is based on Gabor wavelet transform. Gabor wavelets are biologically motivated convolution kernels in the shape of plane waves restricted by a Gaussian envelope function. We use the Gabor wavelet because it can extract the human face feature well. The family of Gabor kernels

$$\varphi j(\vec{x}) = \frac{k_j^{\,2}}{\sigma^2} \exp\left(-\frac{k_j^{\,2} x^2}{2\sigma^2}\right)\left[\exp(\, i\vec{k}_j\vec{x}) - \exp\left(\frac{-\sigma^2}{2}\right)\right]$$

in the shape of plane waves with wave vector $\vec{k}_j$, restricted by a Guassian envelope function. We employ a discrete set of 5 different frequencies, index $v = 0, 1,\ldots,7$ and 8 orientations, index $\mu = 0, 1,\ldots,7$

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos \varphi_\mu \\ k_v \cos \varphi_\mu \end{pmatrix}, k_v = 2^{-\frac{v+2}{2}} \pi , \varphi_\mu = \mu \frac{\pi}{8}$$

with index $j = \mu + 8v$, and $\sigma = 2\pi$ .

Gabor wavelet transformation is done by convolution of the image with the 35 Gabor filters shown in figure 5 above. A jet describes a small patch of gray values in an image $T(\vec{x})$ around a given pixel $\vec{x} = (x,y)$. A jet $J$ is defined as the set $\{Ji\}$ of 40 complex coefficients obtained for one image point. It can be written as

$$J_i = a_j \exp(i\phi j)$$

with magnitudes $a_{j(\vec{x})}$, which slowly vary with position, and phase $\phi_{j(\vec{x})}$, which rotate at a rate approximately determined by the spatial frequency or wave vector $\vec{k}_j$ of the kernels. The set of 40 coefficients obtained for one image point is referred as a jet. A collection of this jets, together with the relative location of the jets form an image graph in the right.

The paper suggests two kind of similarity to compare two jets. A simple method is to compare the magnitude of the jet with the amplitude similarity function

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

However, jets taken from image points only a few pixels apart from each other have very different coefficients due to phase rotation. This may decrease the accuracy of matching. Therefore, we have another method to compare the jets. This method takes into account the phase difference in comparison, the phase similarity function

$$S_a(J, J') = \frac{\sum_j a_j a'_j \cos(\phi_j - \phi'_j - \vec{d}\vec{k}_j)}{\sqrt{\sum_j a_j^2 \sum_j a'_j{}^2}},$$

Using this phase function, the phase difference $(\phi_j - \phi'_j)$ is compensated by the displacement $\vec{d}$ , which is estimated using Taylor expansion. The displacement estimation could be done using the disparity estimation. (FLEET & JEPSON, 1990; THEIMER & MALLOT, 1994).

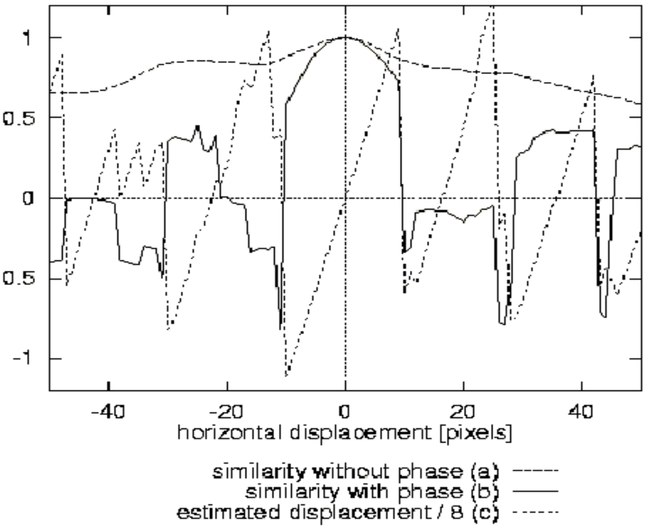

Figure 3 Phase similarity across a horizontal line of a face.

Figure 3 above shows the difference of two similarity functions and the displacement found. Line (a) represents the amplitude similarity and line (b) represents the phase similarity. This line measures the similarity of the right eye and the left eye of a face. Left eye positioned at 0 pixels, while right eye positioned at –24 pixels. From the figure, we can see that we cannot accurately locate the position of right eye by amplitude similarity. With the phase similarity together with estimated displacement, we can accurately locate the right eye for which line (b) is at maximum and displacement is zero.

To represent a face, we need to build an image graph from a set of fiducial points like the pupils, the corner of the mouth, the tip of the nose, the top and bottom of ears, etc. A labeled graph *G* representing a face consists of *N* nodes on the fiducial points at position $\vec{x}_n$ , $n = 1, ...,N$ and *E* edges between them. An image graph is shown in right side of Figure 6, which looks like a grid. For this image graph, 9 fiducial points are used as nodes.

For an automatic face recognition system, it has to locate the fiducial point and build the image graph from an input image automatically. This can be done by matching the input image with a stack like general representation of faces, Face Bunch Graph (FBG). A FBG consists of bunches, which are sets of jets of wide range variation of appearance of a face. Figure 8 shows a face bunch graph. There are set of jets in a node (a bunch) to represent a fiducial point, each with different variations. For example, the eye bunch may consist of jets of open eye, closed eye, male and female eye. With the variations, people with different facial expression could be matched accordingly.

In order to accurately and efficiently locate the fiducial points of an image, two types of FBG are used at two different stages. At normalization stage, a face position is found from an image, a FBG of 30 different models are used. At graph extraction stage, fiducial points are accurately found to build an image graph of the image. This requires FBG of larger size including 70 different models to match accurately.

For the matching between an input graph and the FBG, a function called graph similarity is employed. This function depends on the jet similarity mentioned before and the distortion of the image grid relative to the FBG grid. For an image graph $g^{I}$ with nodes $n = 1,\dots,N$ and edges $e = 1,\dots,E$ and an FBG B with model graphs $m = 1,\dots,M$ . The similarity is defined as

$$S_B(g^I, B) = \frac{1}{N}\sum_n \max_m \left(S_\phi(J_n^I, J_n^{B_m})\right) - \frac{\lambda}{E}\sum_e \frac{(\Delta\vec{x}_e^{\,I} - \Delta\vec{x}_e^{\,B})^2}{(\Delta\vec{x}_e^{\,B})^2}$$

where $\lambda$ determines the relative importance of jets and metric structure. $J_n$ are the jets at nodes *n*, and $\Delta\vec{x}_e$ are the distance vectors used as labels at edge *e*.

In order to extract the image graph from an image, two main steps of matching are needed. The first step is to find the location of a face from the image by using the smaller size FBG. This step is further divided into 3 sub-steps. The first one is to find the approximate face position. The second one is to refine the position and size of the grid found. The last sub-step is to further refine the size of the grid and find the aspect ratio of the face, i.e. the grid. We could then accurately locate the position of a face in the image after applying these steps. After that, step two is performed to find the local distortion of the grid. This helps us finding the fiducial points inside the grid accurately with the use of larger size FBG.

We first perform a wavelet transform using the Gabor filters. The amplitude of the jets is then extracted. After that, we apply the two steps mentioned before. We find the face from the image using the normalization stage FBG. A grid locating the face position is found. Finally, we use the graph extraction stage FBG to get the distorted grid by using local distortion. An image graph will be extracted from the image after going through all the processes.

To recognize a image, we simply compare the image graph to all modal graph and pick the one with the highest similarity value. The similarity function is an average over the similarities between pairs of corresponding jets. If $g^I$ is the image graph, $g^M$ is the modal graph, and node $n_{n'}$ is the modal graph corresponds to node *n'* in the image graph, the define graph similarity is

$$S_g(gI, gM) = \frac{1}{N'}\sum_{n'} S_a(J_{n'}^{I}, J_{n_{n'}}^{M})$$

where the sum runs only over the *N'* nodes in the image graph with a corresponding node in the modal graph.

## COMPARISON

After discussing the above three algorithms, we would like to make a comparison on the advantages and Disadvantages of the methods (see Table I). We found that all three methods are based on statistical approach. They work by extracting the 2D face features from the 2D images. Eigenface and Fisherface find face space based on the common face features of the training set images. Elastic Bunch Graph Matching take local face features like eye, mouth into account for recognition.

Eigenface and Fisherface are global approach of face recognition which takes entire image as a 2-D array of pixels. Both methods are quite similar as Fisherface is a modified version of eigenface. Both make use of linear projection of the images into a face space, which take the common features of face and find a suitable orthonormal basis for the projection. The difference between them is the method of projection is different; Eigenface uses PCA while Fisherface uses FLD. PCA works better with dimension reduction and FLD works better for classification of different classes.

In Elastic Bunch Graph Recognition is based on the fiducial points of an image but not the entire image like Eigenface and Fisherface. This is more suitable for face recognition because it provides important features from the face.

### *Eigen face*

Eigenface is a practical approach for face recognition. Due to the simplicity of its algorithm, we could implement an Eigenface recognition system easily. Besides, it is efficient in processing time and storage. PCA reduces the dimension size of an image greatly in a short period of time. The accuracy of Eigenface is also satisfactory (over 90 %) with frontal faces.

However, as there has a high correlation between the training data and the recognition data. The accuracy of Eigenface depends on many things. As it takes the pixel value as comparison for the projection, the accuracy would decrease with varying light intensity. Besides, scale and orientation of an image will affect the accuracy greatly. Preprocessing of image is required in order to achieve satisfactory result  Advantages of this algorithm are that the eigentfaces were invented exactely for that purpose what makes the system very efficient. A drawback is that it is very sensitive for lightening conditions and the position of the head, it Fast on Recognition, and Easy to implement

Disadvantages-Finding the eigenvectors and eigenvalues are time consuming on PPC The size and location of each face image must remain similar PCA (Eigenface) approach maps features to principle subspaces that contains most energy

### *Fisher face*

Fisherface is similar to Eigenface but with improvement in better classification of different classes image. With FLD, we could classify the training set to deal with different people and different facial expression. We could have better accuracy in facial expression than Eigen face approach. Besides, Fisherface removes the first three principal components which is responsible for light intensity changes, it is more invariant to light intensity.

Fisherface is more complex than Eigenface in finding the projection of face space. Calculation of ratio of between-class scatter to within-class scatter requires a lot of processing time. Besides, due to the need of better classification, the dimension of projection in face space is not as compact as Eigenface, results in larger storage of the face and more processing time in recognition.

- Fisher linear discriminating (FLD, Fisherface) approach maps the feature to subspaces that most separate the two classes.

### *Elastic Bunch Graph Matching*

*Elastic graph matching* is the basic process to compare graphs with images and to generate new graphs. In its simplest version a single labeled graph is matched onto an image. A labeled graph has a set of jets arranged in a particular spatial order. A corresponding set of jets can be selected from the Gabor-wavelet transform of the image.

The image jets initially have the same relative spatial arrangement as the graph jets, and each image jet corresponds to one graph jet. The similarity of the graph with the image then is simply the average jet similarity between image and graph jets.In order to increase the similarity one allows the graph to translate, scale and distord to some extent, resulting in a different selection of image jets. The distortion and scaling is limited by a penalty term in the matching cost function. The image jet selection which leads to the highest similarity with the graph is used to generate a new graph.When a bunch graph is

used for matching, the procedure gets only a little bit more complicated. Beside selecting different image locations the graph similarity is also maximized by selecting the best fitting jet in each bunch. This is done independently of the other bunches to take full advantage of the combinatorics of the bunch graph representation.This algorithm takes advantage of the fact that all human faces share a similar topological structure. This makes it possible to represent the face as a labeled graph. The nodes and edges of the graph contain additional information as for example the distance from one node to another. In contrast to the Eigeface algorithm the Elastic Bunch Graph Matching treats one vector per feature of the face. A feature for the face are the eyes, nose, mouth etc. This has the advantage that changes in one feature (p.e. eyes open, closed) do not necessarily mean that the person is not recognized any more. In addition this algorithm makes it possible to recognise faces up to a rotation of 22 degres. Drawbacks of this algorithm are that it is very sensitive to lightening conditions and that a lot of graphs have to be placed manually on the face. For a reliable system one needsrecognition is then done by comparing this graphes, it requires huge storage of convolution images for better performance.

Eigenface is essentially a technique that using the minimum distance classifier, which is optimal if the lighting variation between the training set and recognition set can be modeled as zero-mean. It also success when the mean is nonzero but small. When the changes in lighting are large, the result will have a significant decrease in the recognition rate. The reason is that the distance between two face images is dominated by the lighting factor rather than the differences between the two faces. If the pose is varied much, the training set need to have other profiles view in order to recognize such poses. If the eigenface is used in a practical system, the scale, position and lighting conditions should be provided for the system to ensure high recognition rate. Eigenface can take the advantages of computational efficiency when the eigenfaces are stored and the dimension of these vectors is not large.

Elastic Bunch Graph Matching make use of Gabor features, being the output of bandpass filters, and this are closely related to derivatives and are therefore less sensitive to lighting change. Also, this approach uses features only at a key node of the image rather than the whole image, this can reduce the noise taken from the background of the face images. Together with other important advantages of it is that it is relatively insensitive to variations in face position, facial expression. The matching procedure uses the FBG as a face template for finding the precise fiducial point, which solve the problem for automatically localization. The stored data can be easily expanded to a database for storage. When a new face images is added, no additional afford is needed to modify templates, as it already stored in the database. This advantage had overcome the eigenfaces because the eigenface need to be recalculated.

## CONCLUSION

In this paper, we have discussed the problems for face recognition such as light intensity variable, facial expression etc. We have checked theoretical as well as statistical aspect of the three different statistical approach face recognition algorithm (Eigenface, Fisherface and Elastic Bunch Graph Matching). Finally, we have made a comparison of these algorithms and have discussed the advantages and Disadvantages of each of them. eigenface and elastic bunch graph matching. For the elastic bunch graph matching, it makes use of Gabor feature, which are insensitive to lighting variation, rigid, and

deformable matching. This allows for the position and facial expression variation, because features only taken from a key points in the face image rather than the whole images. about 70 graphs. With this algorithms the face is represented as a graph. The nodes of the graph are positionet at special places as the nose, eyes etc. The edges between the nodes are labeled with 2D distance vectors. The nodes are labeled with socalled “jets”. This is a set of Gabor wavelet coefficients at different scales and orientations. Face.

## REFERENCES

[1] M. A. Turk and A. P. Pentland, “Face Recognition Using Eigenfaces”, Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, June 1991.

[2] Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D.J. “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection”, Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 19, pp 711-720, Issue: 7, July 1997

[3] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg, “Face Recognition by Elastic Bunch Graph Matching”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 19, pp. 775-779, No.7 July 1997

[4] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg, “Face Recognition by Elastic Bunch Graph Matching”, “Intelligent Biometric Techniques in Fingerprint and Face Recognition”, eds, L.C . Jain et al., publ. CRC Press, ISBN 0-8493-2055-0, Chapter 11, pp. 355-396, (1999)

[5] Jun Zhang, Yong Yan, and Martin Lades, “Face Recognition: Eigenface, Elastic Matching, and Neural Nets”, Proceedings of the IEEE, VOL. 85, NO. 9, September 1997

[6] Tai Sing Lee, “Image Representation Using 2D Gabor Wavelets”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 18, No. 10, October 1996

[7] Yao Hongxun; Gao Wen; Liu Mingbao; Zhao Lizhuang, “Eigen features technique and its application”, Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on , Volume: 2 , 2000 Page(s): 1153 -1158 vol.2

[8] Lyons, M.J.; Budynek, J.; Akamatsu, S. “Automatic Classification of Single Facial Images”, Pattern

Analysis and Machine Intelligence, IEEE Transactions on , Volume: 21 Issue: 12 , Dec. 1999 , Page(s): 1357 -1362

[9] Liu, C.; Wechsler, H., Evolutionary pursuit and its application to face recognition", Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 22 Issue: 6 , June 2000 , Page(s): 570 –582

[10] Lades, M.; Vorbruggen, J.C.; Buhmann, J.; Lange, J.; von der Malsburg, C.; Wurtz, R.P.; Konen, W. "Distortion invariant object recognition in the dynamic link architecture", Computers, IEEE Transactions on , Volume: 42 Issue: 3 , March 1993 , Page(s): 300 –311

[11] Moghaddam, B.; Wahid, W.; Pentland, A." Beyond eigenfaces: probabilistic matching for face recognition", Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on , 1998 ,Page(s): 30 -35.