

**Face Recognition
Algorithms Review
Term Paper - December 2001**

Tang Ho Man, Sunny
Email: hmtang@cse.cuhk.edu.hk

Supervised by
Prof. Michael Lyu

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong

Abstract

In this paper, we look into an important field of biometrics, face recognition. We first discuss the problems and requirements of a face recognition system. Then, we review three face recognition algorithms, Eigenfaces, Fisherfaces and Elastic Bunch Graph Matching, and make a comparison of the advantages and drawbacks of each algorithm.

1. Introduction

The study of biometrics is becoming important in recent years. Several security applications are developed based on biometric personal identification such as computerized access control. With personal identification, identity of a person can be determined, preventing unauthorized access of important data. Several biometrics signals are used for this kind of application, face recognition, speech, iris, fingerprint, signatures, are instances. Within these signals, face recognition would be addressed here due to its widely usage in the field of security application and multimedia search engines.

Face recognition provides us a convenient way to identify and recognize a person in a large database. With face recognition, we can recognize a person by just taking a photo of that person. User no longer needs to scan his fingerprint or iris for personal identification but just need to stand in front of a camera. The system can check its database to recognize the person from his image.

Apart from the convenience face recognition provides, it can be applied in multimedia search engine. Fast growing on multimedia technology and Internet technology enables searching for multimedia data like video clips possible. However, information retrieval within vast amount of multimedia data is still a challenging task. With face recognition and video segmentation technology, we can find video clips of a particular person easily by simply supply with the search engine a picture of that person. All related video like news clips would be found.

In the following parts of this paper, we would discuss the important problems and requirements for a face recognition system. We would address the problems we may face and the requirement we should meet for implementing a reliable face recognition system. Afterwards, we would describe three kinds of face recognition algorithms, namely Eigenface, Fisherface and Elastic Bunch Graph Matching. And then make a comparison and discuss the advantages and drawbacks of each of these.

2. Problems and Requirements

2.1. Problems

An automated face recognition system needs to overcome several problems. One of the big problems is the ability to identify a person whose picture is not taken straight on. That means the face may not be frontal. It is not easy to make a system capable to recognize a person with a rotated face. Besides, size of the image would affect the recognition result because some approach requires a standard size images. And small size image makes the resolution of the image not clear enough for recognition. Another problem for face recognition is an appearance of a person may change drastically over a short period of time. For examples, day-to-day facial differences due to glasses, makeup and head hair style. All these changes may face recognition of a person difficult.

Apart from these, lighting condition is another major problem for face recognition. The same person under different lighting condition may be seen quite different. As shown in figure 1, 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.



Figure 1. In the left image, the dominant light source is nearly head-on; In the right image, the dominant light source is from above and to the right.

2.2. Requirements

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 80%. Otherwise, we cannot correctly recognize a person. Efficiency is critical for a real-time face recognition system. The processing time for an input image should be within 1 minute. 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.

3. 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 use linear projection while the third algorithm Elastic Bunch Graph Matching uses graph and wavelet transformation to recognize a face.

3.1. Eigenface

Eigenface was suggested by Alex. P. Pentland and Matthew A. Turk of MIT in 1991. 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 128 x 128 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. ($M \ll N^2$). As M is much smaller than N^2 , comparison between vectors would be efficient.

PCA is done by first finding the average face ψ by averaging the training set images $\{T_1, T_2, \dots, T_M\}$ with T_i representing each of the vector in the set. Then we form a matrix $A = \{\phi_1, \phi_2, \dots, \phi_M\}$ with column vector $\phi_i = T_i - \psi$, which is the difference vector of the train images and the average face. We can then get the covariance matrix $C = AA^T$ and the eigenvector and the associated eigenvalues of C .

After the eigenvectors have been calculated, the eigenvalues of each eigenvector are sorted. These vectors are known as eigenfaces. The eigenfaces with the largest number of eigenvalues are chosen. These M' (where $M' < M$) eigenfaces are considered the best eigenvector to represent a face. The span of the M' eigenfaces are called face space. Figure 2 below shown a few of low order eigenfaces used for projection.



Figure 2. Standard eigenfaces

Second phase of this algorithm is recognition phase. In this phase, a new image is obtained. To recognize this image, we first subtracted the image by the average face ψ . Then we calculate the dot product of the input vectors with the eigenfaces. This makes a projection of the input image onto the face space. Similarly, we make projections of the training image onto the face space. Figure 3 shows the projection of image onto the face space, which appears as the point in the plane. The euclidean distances of point of the input image with the points of training set are then computed. The training set image with minimum distance from the input image should be the best match.

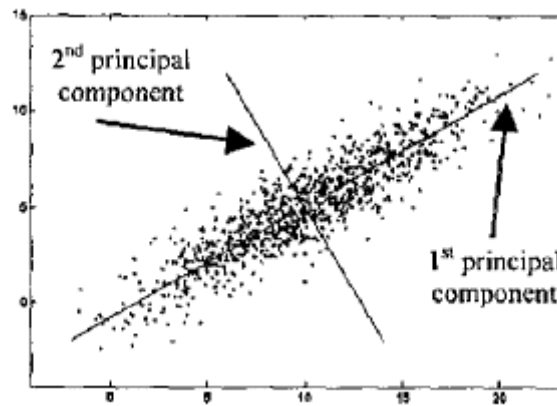


Figure 3. Examples of principal components analysis in a 2-D distribution of data.

However, there may be cases that the input image is not in the training set. This would still find a best match of the input image, but this best match is not the correct one. Therefore, we can set a distance threshold for the recognition by trial and error until a satisfactory one is found. When the minimum distance found is larger than the threshold, we can regard the input image is not in the training set.

In the experiment the effects of varying lighting, size and head orientation were investigated using a database of 2500 images. Experiment result shows that eigenface approach reach 96% correct classification averaged over lighting variation, 85% correct averaged over orientation variation and 64% correct averaged over size variation.

3.2. Fisherface

Fisherface was suggested by Peter N. Belhumeur, Joao P. Hespanha and David J. Kriegman of Yale Univeristy in 1997. This approach is similar to eigenface approach, which makes use of projection of image into a face space, with improvements on insensitive to large variation in lighting and facial expression.

Eigenface method uses PCA for dimensionality reduction, which yields projection directions that maximize the total scatter across all classes of images. This projection is best for reconstruction of images from a low dimensional basis. However, this method doesn't make use of between-class scatter. The projection may not be optimal from discrimination for different classes. Let the total scatter matrix S_T be defined as

$$S_T = \sum_{k=1}^N (T_k - \psi)(T_k - \psi)^T$$

The projection W_{opt} is chosen to maximize the determinant of the total scatter matrix of the projection sample, i.e.

$$\begin{aligned} W_{opt} &= \arg \max_W |W^T S_T W| \\ &= [w_1, w_2, \dots, w_m] \end{aligned}$$

where $\{w_i | i=1, 2, \dots, m\}$ is the set of n -dimensional eigenvectors of S_T corresponding to the m largest eigenvalues.

Fisherface method uses Fisher's Linear Discriminant (FLD) by R.A. Fisher. This projection maximizes the ratio of between-class scatter to that of within-class scatter. The idea is that it tries to "shape" the scatter in order to make it more reliable for classification. Let the between-class scatter matrix be defined as

$$S_B = \sum_{l=1}^C N_l (\psi_l - \psi)(\psi_l - \psi)^T$$

and the within-class scatter matrix be defined as

$$S_W = \sum_{l=1}^C \sum_{T_k \in T_l} (T_k - \psi_l)(T_k - \psi_l)^T$$

where ψ_l is the mean image of class T_l . The optimal projection W_{opt} is chosen as the matrix with orthonormal columns, which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples, i.e.

$$\begin{aligned} W_{opt} &= \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|} \\ &= [w_1, w_2, \dots, w_m] \end{aligned}$$

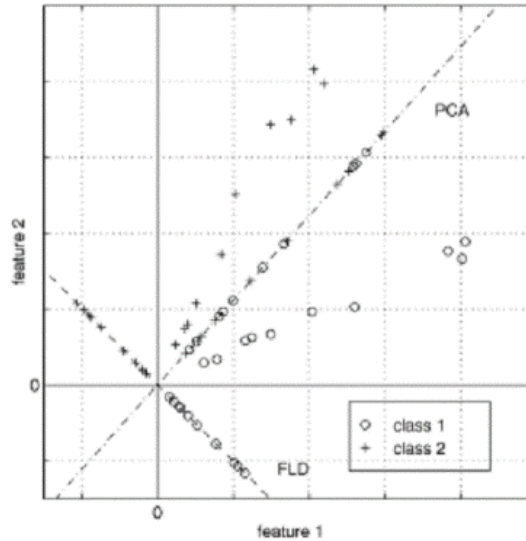


Figure 4. A comparison of principal component analysis (PCA) and Fisher's linear discriminant (FLD) for a two-class problem where data for each class lies near a linear subspace.

Besides, this method projects away variation in lighting and facial expression while maintaining discriminability. For lighting variation, the variation due to lighting is reduced by discarding the three most significant principal components. This is because the first three principal components contribute the lighting variations. This results in better performance under variable lighting conditions. For facial expression variation, we can divided the training images into classes based on the facial expression. Take glasses recognition as an example, the training set can be divided into two main classes: "wearing glasses" and "not wearing glasses". With this set of training data, Fisherface can correctly recognized people even he is wearing glasses. Therefore, Fisherface works well with variation in lighting and facial expression.

Experiments are conducted to compare the error rate of two approaches mentioned, Eigenface and Fisherface using Yale face database which contains variation in facial expression and lighting. Table 1. below shows the result:

Face Recognition Method	Error Rate (%)	
	Close Crop	Full Face
Eigenface	24.4	19.4
Eigenface w/o first 3 principal components	15.3	10.8
Fisherface	7.3	0.6

Table 1. The relative performance of algorithms under Yale database.

3.3. Elastic Bunch Graph Matching

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.

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

$$\phi_j(\bar{x}) = \frac{k_j^2}{\sigma^2} \exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right) \left[\exp(i\bar{k}_j \bar{x}) - \exp\left(-\frac{\sigma^2}{2}\right) \right]$$

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

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

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

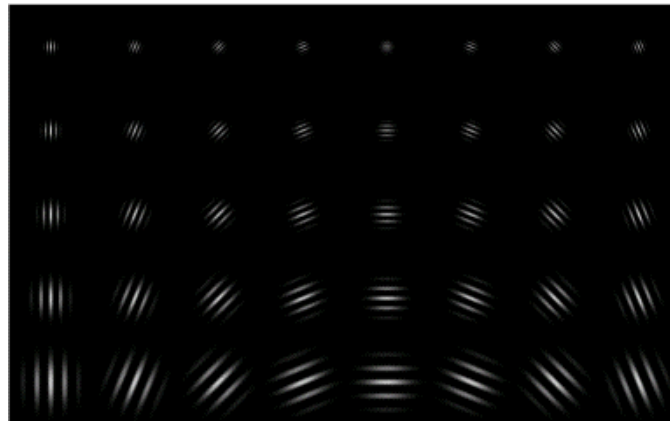


Figure 5. Gabor filter of 5 frequencies and 8 orientations.
From high frequencies to low frequencies.

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

$$J_i = a_j \exp(i\phi_j)$$

with magnitudes $a_{j(\bar{x})}$, which slowly vary with position, and phase $\phi_{j(\bar{x})}$, which rotate at a rate approximately determined by the spatial frequency or wave vector $\bar{k}_j$ of the kernels. Figure 6 below shows a convolution is made between the original image and the Gabor wavelets. 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.

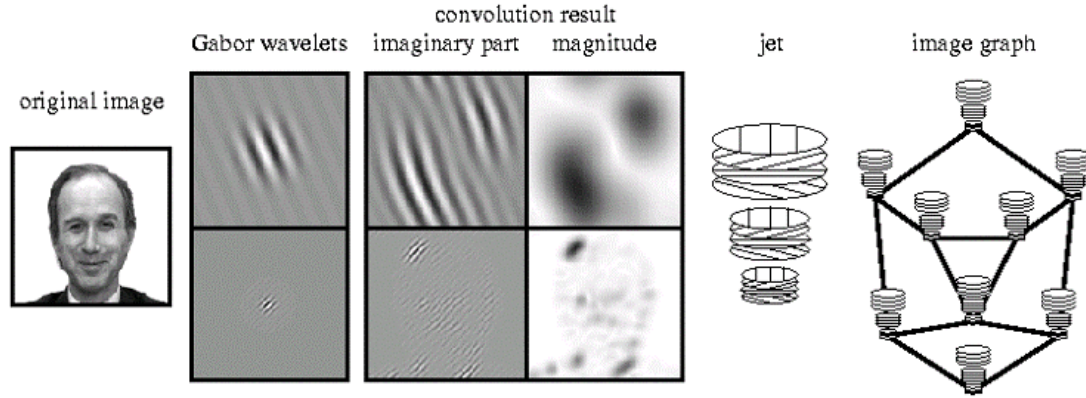


Figure 6. Convolution of an image and Gabor wavelets, jet of a point, image graph of the face.

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

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

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

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

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

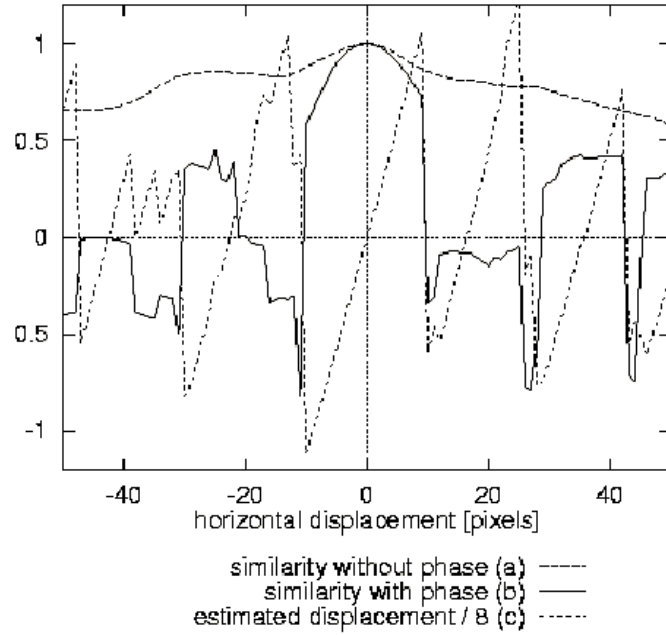


Figure 7. Phase similarity across a horizontal line of a face.

Figure 7 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 $\bar{x}_n$, $n = 1, \dots, N$ 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.

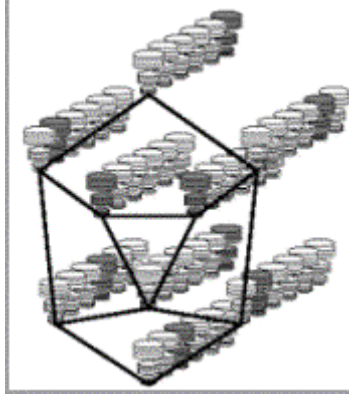


Figure 8. Face bunch graph.

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

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

where λ determines the relative importance of jets and metric structure. J_n are the jets at nodes n , and $\Delta \bar{x}_e$ 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.

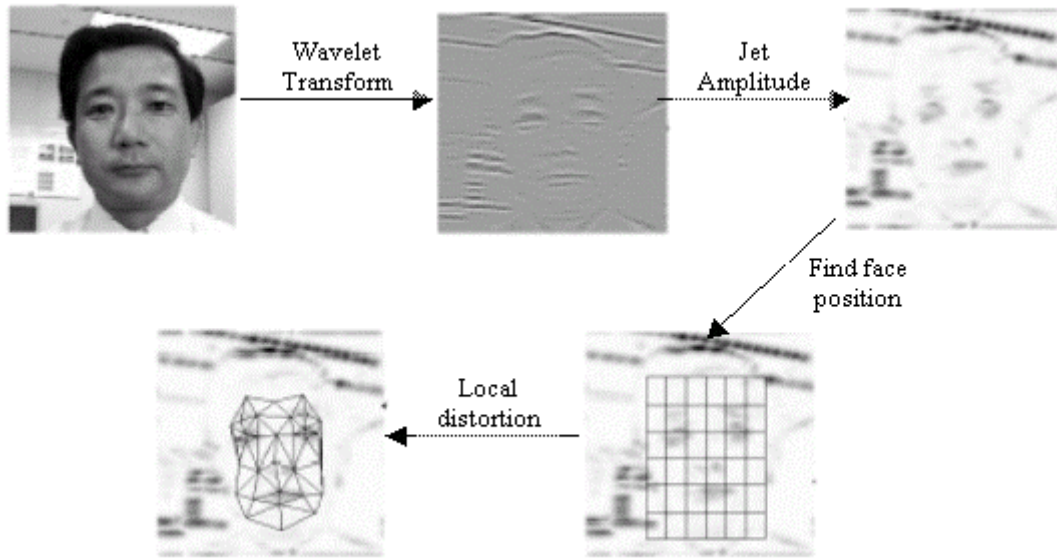


Figure 9. Overall steps for graph extraction

Figure 9 shows the overall step of graph extraction from an image. 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

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

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

Experiment is done using Bochum database to test for recognition of rotated face against frontal face with variation in facial expression. Result shows that Elastic Bunch Graph Matching achieves 91% accuracy with frontal view, 94% accuracy with rotation of 11 degree, 88% accuracy with rotation of 22 degree. Notice that the accuracy for 11 degree rotated is higher than that of frontal; this indicates that the variation due to facial expression is relatively larger than face rotation.

4. Comparison of Advantage and Drawback

After reviewing the above three algorithms, we would like to make a comparison on the advantages and drawbacks of each of them. We found that all three methods are based on statistical approach. They work by extracting the face features from the 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.

Elastic Bunch Graph Matching is a local approach of face recognition. 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 extracts the important features from the face as criteria. Besides, the use of Gabor wavelet is also suitable for human feature extraction because the wavelet is similar to eyes, eye bows etc. By taking convolution of the image with different Gabor wavelets in terms of frequencies and orientation, human feature would be extracted accurately.

4.1. Eigenface

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.

4.2. Fisherface

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 Eigenface 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.

4.3. Elastic Bunch Graph Matching

Elastic Bunch Graph Matching works well with different facial expression. Making use of the general representation of FBG, we can recognize people of different facial expression accurately. Besides, scaling of image is solved at the normalization stage of the algorithm. It can recognize image with different scales. It is also capable of recognizing faces of different pose due to the use of Elastic Bunch Graph. It is invariant to light intensity too.

However, this algorithm has certain drawbacks. It is quite complicated to build the FBG at the initial stage. A large amount of grid placements has to be done manually at the beginning. Besides, it is difficult to implement because of the complexity of the algorithm in automatically finding the position of the fiducial points. And it requires huge storage of convolution images for better performance.

5. Conclusion

In this paper, we have addressed the problems needed to overcome for face recognition such as light intensity variable, facial expression etc. And we have discussed certain requirements for a reliable and efficient face recognition system like accuracy, efficiency. We have reviewed 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 drawbacks of each of them.

6. Acknowledgements

We would like to thank Prof. Michael Lyu and Prof. Irwin King for providing constructive comments and suggestion. The directions and ideas they given are valuable for our research.

7. Reference

- [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