FACE RECOGNITION BY EIGENFACE AND ELASTIC BUNCH GRAPH MATCHING

Master of Philosophy
Research Project First-term Report

SUPERVISED BY

Professor LYU, Rung Tsong Michael

PREPARED BY

JANG KIM FUNG (01036550)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
THE CHINESE UNIVERSITY OF HONG KONG

Table of Content

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 two most recently proposed methods for face recognition. One of the approach is eigenface and other one is the elastic bunch graph matching. After the implementation of the above two methods, we learn the pros and cons of each approach and the difficulties for the implementation.

I. Introduction

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 two techniques are eigenface by Alex P. Pentland [1] and elastic bunch graph matching by Laurenz Wiskott [2]. These techniques are recent and have apparently promising performances, and are representative of new trends in face recognition.

Section II provides an overview of eigenface. Those descriptions for elastic bunch graph will be in section III. Section IV gives the experimental results and some theoretical analysis of their strengths and limitations, and followed by a summary in Section V. The last section will have the future work for the study.

II. Eigenface

In this section, we will discuss the eigenface included idea, algorithm and theoretical analysis. Eigenface is a global approach of face recognition which treats face image as a two-dimensional recognition problem, and assume that face are upright at normal to have a smaller set of two-dimensional characteristic views.

First of all, we wish to find the principal components of the faces or the eigenvectors of the covariance matrix of the set of face images. These eigenvector may be a set of features which can be represent the face images. The eigenvector can be displayed as a ghostly face which we call an eigenface. Some of these faces are shown in Figure 1.

Figure 1 Examples of eigenface

The number of eigenface is equal to the number of face image in the training set. However, the eigenface can be approximated by using those have the largest eigenvalues within the training set images. Computational efficiency is the reason why we need to use fewer eigenfaces. The best *M'* eigenfaces span an *M'* dimensional subspace, called face space.

By using the principal component analysis developed by Sirovich and Kirby [4], face can be efficiently represented, and argued that a collection of face images can be approximately reconstructed by storing a small collection of weights for each face and a small set of standard images.

Figure 2 Original images

Figure 3 Reconstructed images

A. Calculating Eigenfaces

Let a face image *I(x, y)* be a two-dimensional *N* by *N* array of intensity values. Images of faces will not be randomly distributed in this huge space (N^2) and we can use a relatively low dimensional subspace to describe it. Use the PCA to find out the vectors which best account for the distribution of face images. These vectors are defined as "face space" and each vector is of length N^2. Because these vectors are eigenvector of the covariance matrix corresponding to the original face images, also the appearance like a face, we call them as "eigenfaces".

Let the training set of face images be $\Gamma_1, \Gamma_2, \Gamma_3 ... \Gamma_M$. The average face of the training set will be

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n \tag{1}$$

The face difference is

$$\Phi_i = \Gamma_i - \Psi \tag{2}$$

These set of large vectors is then performed principal component analysis to find the *M* orthonormal vectors μ_n and their associated eigenvalues λ_k, the covariance matrix

$$\begin{aligned} C &= \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \\ &= AA^T \end{aligned} \tag{3}$$

where matrix $A = [\Phi_1 \Phi_2 ... \Phi_M]$, The matrix C is N^2 by N^2, determining the N^2 eigenvectors and eigenvalues is infeasible. However, we can solve is by first calculate a much smaller *M* by *M* matrix problem, and taking linear combinations of the resulting vectors. These greatly reduce the computational time 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 our practice, the training set of face image will be relatively small when compare with the number of pixels ($M << N^2$). The associated eigenvalues can then be used to rank the eigenvectors to extract the most useful ones.

B. Classify a face image

After created the eigenfaces, identification can be started. The eigenfaces span an *M'*-dimensional subspace of the original N^2 image space. The *M'* significant eigenvectors of the L matrix are selected by the largest associated eigenvalues. In practice, *M'* = 7 eigenfaces were used when *M* = 16 face images.

A new face image (Γ) is projected into the "face space" to form weights

$$\varpi_k = \mu_k^T (\Gamma - \Psi) \tag{4}$$

where $k = 1, \ldots, M'$.

The weights form a vector

$$\Omega^T = [\varpi_1 \varpi_2 ... \varpi_{M'}] \tag{5}$$

that describes the contribution of each eigenfaces in representing the input face image (Γ), eigenfaces used as a basis set for face images. The simplest method to determine which face class gives the best description of an input face image is to find the face class *k* that minimizes the Euclidian distance

$$\varepsilon_k = \| (\Omega - \Omega_k) \| \tag{6}$$

where Ω_k is a vector describing the k^{th} face class. A face is classified as belonging to class *k* when the minimum ε_k is below some chosen threshold θ_ε. Otherwise the face is classified as "unknown".

C. Face space revisited

An image of face should lie near the face space, which we call "face-like". That mean the projection distance ε should be less than the threshold θ_δ. A known face image should be project to near the corresponding face class, ($\varepsilon_k < \theta_\varepsilon$). These illustrate four cases:

	Face space	Face class
Case 1	Near	Near
Case 2	Near	Not near
Case 3	Distant	Near
Case 4	Distant	Not near

Table 1 Four cases happened in recognition

In the first case, face image is recognized and identified. In second case, an unknown face is present. The last two cases indicate that the image is not a face. Case three also mean

that a false positive in the system. This is due to the significant distance between the image and the subspace of expected face images.

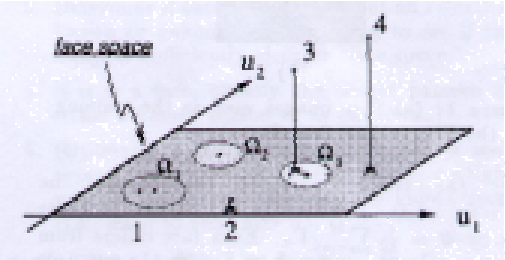

Figure 4 Face space to illustrate the four cases of projecting an image into face space. In this case, there are two eigenfaces ($\mu_1 and \mu_2$) and three known face class ($\Omega_1, \Omega_2 and \Omega_3$)

III. Elastic Bunch Graph Matching

This approach has used the structure information of a face which reflects the fact that the images of the same subjects trend to translate, scale, rotate, and deform in the image plane. It makes use of the labeled graph, edges are labeled the distance information and nodes 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. This can make the system robust to large variation in the images.

A. Preprocessing using Gabor Wavelets

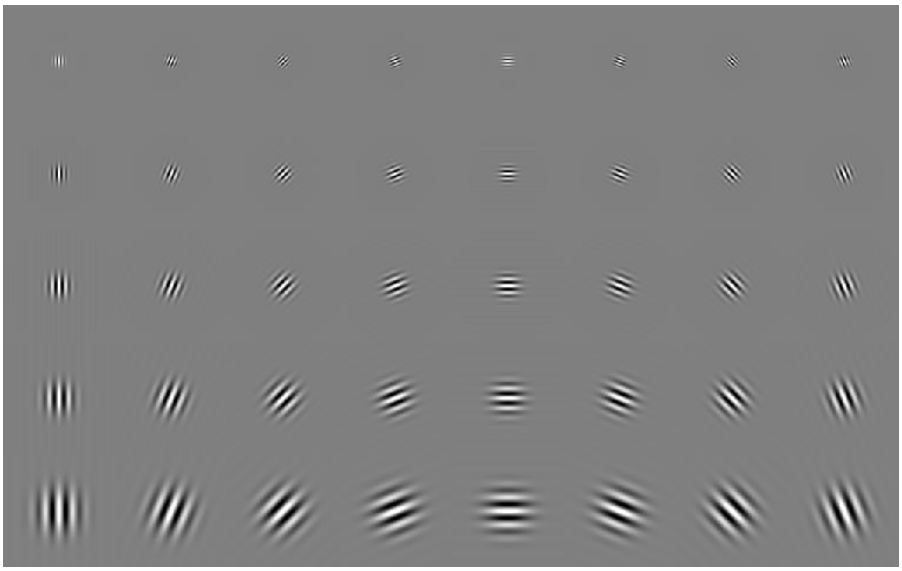

Figure 5 The real part of the Gabor filter with 5 frequencies and 8 orientations

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 plane waves restricted by a Gaussian envelope function [5]. The set of convolution coefficients for kernels of different orientations and frequencies at one image pixel is called a jet.

A.1. Jets

A jet means a set of gray values in an image $I(\vec{x})$ around a given pixel $\vec{x} = (x, y)$, which is based on a wavelet transform, defined as a convolution

$$J_j(\vec{x}) = \int I(\vec{x}')\psi_j(\vec{x} - \vec{x}')d^2\vec{x}' \tag{7}$$

with a family of Gabor kernels

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2}\exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right)\left[\exp\left(i\vec{k}_j\vec{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \tag{8}$$

in the shape of plane waves with wave vector $\vec{k}_j$, the function is restricted by a Gaussian envelope function - $\exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right)$. Using 5 different frequencies, index $\nu = 0,...,4$, and 8 orientations, index $\mu = 0,...,7$,

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_\nu \cos\varphi_u \\ k_\nu \sin\varphi_u \end{pmatrix}, k_\nu = 2^{-\frac{\nu+2}{2}}\pi, \varphi_u = \mu\frac{\pi}{8}, \tag{9}$$

with index $j = \mu + 8\nu$. The width σ / k of the Gaussian is controlled by the parameter $\sigma = 2\pi$. The second term in the bracket of Eq. (8) makes the kernels DC-free, i.e., the integral $\int \psi_j(\vec{x})d^2\vec{x}$ vanishes. All kernels are generated from one mother wavelet by dilation and rotation because the family of kernels is self-similar. Example of the Gabor filter we used is shown in Figure 5.

A jet J is defined as the set $\{J_i\}$ of 40 complex coefficients for one image points

$$J_i = a_j \exp\left(i\phi_j\right) \tag{10}$$

where a_j is the magnitudes $a_j(\vec{x})$, which slowly vary with position, and phases $\phi_j(\vec{x})$, which rotate at a rate approximately determined by the spatial frequency or wave vector $\vec{k}_j$ of the kernels;

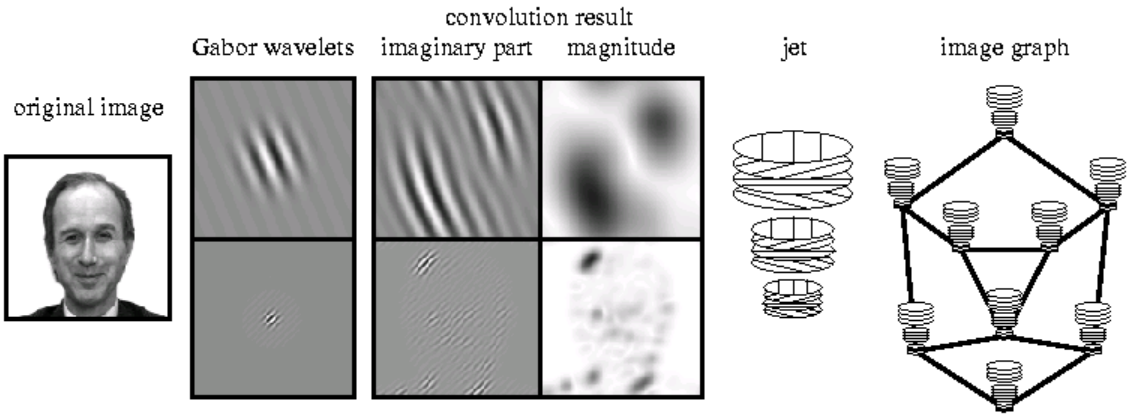

Figure 6 The graph representation of a face is based on the Gabor wavelet transform, a convolution with a set of wavelet kernels.

The face image is first convolution with the Gabor filter, the result is shown in Figure 6, then continue to have a convolution with the 5x8 Gabor wavelet, the set of 40 coefficients obtained for one image point form a jet. A sparse collection of such jets together with distant information of relative location constitutes an image graph. In Figure 6, only 3 frequencies and 4 orientations are used and 9 nodes are used to form an image graph.

Gabor wavelets were chosen to use because of their robustness and biological relevance. Normalizing the jets can provide robustness against varying brightness in the image. The limited localization in space and frequency yields a certain amount of robustness against translation, distortion, rotation, and scaling.

A.2. Comparing Jets

Jets taken from image points only a few pixels apart from each other have very different coefficient due to the characteristic of phase rotation. This can cause problems for matching. Thus we either ignore the phase or compensate for it. The similarity function without phase [6]

$$S_a(J,J') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j^2 \sum_j a'^2_j}}, \tag{11}$$

This is a smooth function with local optima forming large attractor basins, leading to rapid and reliable convergence with simple search methods, for example, stochastic gradient descent.

Pattern between similar magnitudes can be discriminated by using phase information, and the accurate jet localization in an image can also be found because phase varies so quickly with location. Assuming that two jets J and J' refer to locations with small relative displacement $\vec{d}$, the phase shifts can be approximately compensated for by the term $\vec{d}\vec{k}_j$, this form a phase-sensitive similarity function

$$S_\phi(J, J') = \frac{\sum_j a_j a'_j \cos\left(\phi_j - \phi'_j - \vec{d}\vec{k}_j\right)}{\sqrt{\sum_j a_j^2 \sum_j a'^2_j}}. \tag{12}$$

To estimate the displacement $\vec{d}$, we can maximize S_ϕ in its Taylor expansion, we will detailed discuss in the following section. Having the displacement information, we can use it to find the accurate jet location.

A.3. Displacement Estimation

To find the displacement vector $\vec{d} = (d_x, d_y)$, disparity estimation method is used [7], [8].

That is maximization of the similarity S_ϕ in its Taylor expansion

$$S_\phi(J, J') \approx \frac{\sum_j a_j a'_j \left[1 - 0.5\left(\phi_j - \phi'_j - \vec{d}\vec{k}_j\right)^2\right]}{\sqrt{\sum_j a_j^2 \sum_j a'^2_j}}. \tag{13}$$

By setting $\frac{\partial}{\partial d_x} S_\phi = \frac{\partial}{\partial d_y} S_\phi = 0$ and solving for $\vec{d}$ yield

$$\vec{d}(J, J') = \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \frac{1}{\Gamma_{xx}\Gamma_{yy} - \Gamma_{xy}\Gamma_{yx}} \times \begin{pmatrix} \Gamma_{yy} & -\Gamma_{yx} \\ -\Gamma_{xy} & \Gamma_{xx} \end{pmatrix} \begin{pmatrix} \Phi_x \\ \Phi_y \end{pmatrix}, \tag{14}$$

if $\Gamma_{xx}\Gamma_{yy} - \Gamma_{xy}\Gamma_{yx} \neq 0$, with

$$\Phi_x = \sum_j a_j a'_j k_{jx} \left(\phi_j - \phi'_j\right),$$
$$\Gamma_{xy} = \sum_j a_j a'_j k_{jx} k_{jy},$$

and $\Phi_y, \Gamma_{xx}\Gamma_{yx}\Gamma_{yy}$ defined correspondingly. In this function, the phase differences may exist the range of $\pm\pi$, we need to correct it by $\pm 2\pi$. The displacement can be estimated between two jets when they close enough that their Gabor kernels are highly overlapping.

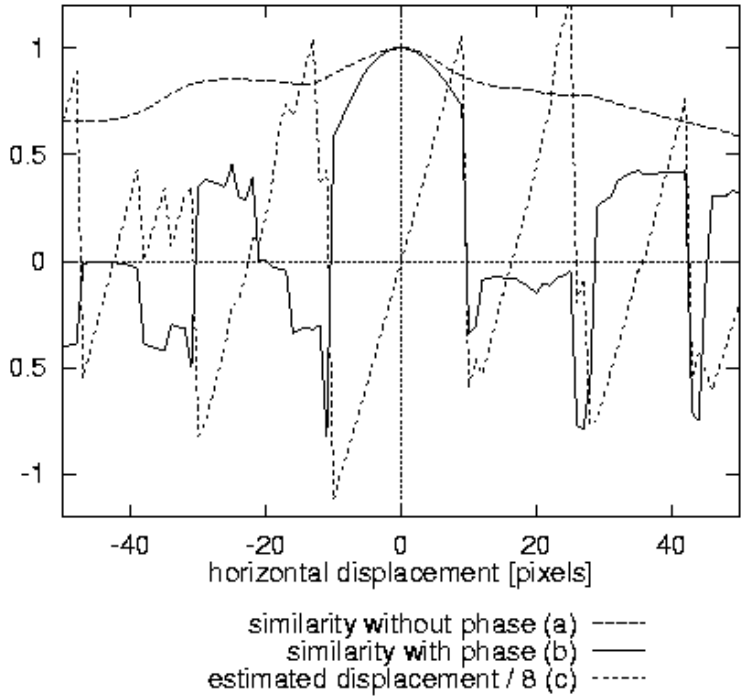

Figure 7 (a) Similarity $S_a\left(J(\vec{x}_1), J'(\vec{x}_0)\right)$ with jet *J'* taken from the left eye of the face shown in Figure 6 and jet *J* taken from the same horizontal line with jet *J'*, $\vec{x}_1 = \vec{x}_o + (d_x, 0), d_x = -50,...,50$. (b) Similarity $S_\phi\left(J(\vec{x}_1), J'(\vec{x}_0)\right)$ and (c) estimated displacement $\vec{d}\left(J(\vec{x}_1), J'(\vec{x}_0)\right)$ for the same jets as in (a) using focus 1.

In Figure 7, the similarity function without phase is a smooth function which in a range of 0.6-1.0, and we can roughly find there is a local maximum around $\vec{d}_x = -24$ as the right eye is 24 pixels away from the left eye. However, we can't locate the jets precisely. That's why we need to use the similarity function with phase. The estimated displacement is periodic due to the frequency of the kernel. Without modifications, the equation S_ϕ can determine displacements up to half the wavelength of the highest

frequency kernel, which would be two pixels for $k_0 = \pi/2$. The estimated range can be increased by using low frequency kernels only.

We refer to the number of frequency levels used for the first displacement estimation as focus. A focus of 1 mean that only the lowest frequency level is used and the estimated range may be up to 8 pixels. In other word, a focus of 5 means that all five levels is eventually used. For each higher level, the phases of the higher frequency coefficients have to be corrected by multiples of 2π to match as closely as possible the expected phase differences inferred from the displacement estimated on the lower frequency level. In the above iterative refinement process, the accurate position of the jets can be found.

B. Face Representation

B.1. Individual Faces

To represent a face, we use a set of fiducial points, e.g. the pupils, the eyebrows, corners of the mouth, the tip of nose, etc. A labeled graph *G* representing a face consists of *N* nodes on these fiducial points at positions $\vec{x}_n, n = 1,..., N$ and *E* edges between them. Each node are labeled with jets J_n. The edges are labeled with distances $\Delta\vec{x}_e = \vec{x}_n - \vec{x}_{n'}, e = 1,..., E$, where edge *e* connects node *n* with *n'*. This face graph is object-adapted, since the nodes are selected from face-specific point. The position of the jets is selected manually, as some of the nodes may be occluded and the distances vary due to rotation in depth.

B.2. Face Bunch Graph

Automatic finding fiducial points in new faces needs a general representation rather than models of the individual faces. A wide range of possible variations in the appearance of faces, like differently shaped eyes, mouths, or noses, different types of beards, variations due to sex, age, etc, should be covered. Combination each feature by a separate graph is not efficient, we use a stack-like structure, called a face bunch graph (FBG), see Figure 8.

Each model has the same grid structure and the nodes refer to the identical fiducial points. A set of jets referring to one fiducial point is called a bunch. During the location of each fiducial point in a face, the best fitting jet, called the local expert, is selected from the bunch by the procedure described in the next section. Thus, any combination of jets in the bunch graph is available for a wide range of variation than the model of individual faces.

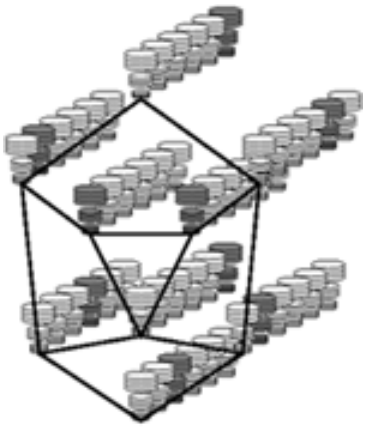

Figure 8 The Face Bunch Graph represent the faces in general.

To form a FBG, assume for a particular pose that there are M model graphs G^{Bm} (m=1,…,M) of identical structure, taken from different model faces. The corresponding FBG B is then given the same structure, its nodes are labeled with bunches of jets J_n^{Bm} and its edges are labeled with the averaged distances

$$\Delta\vec{x}_e^{B} = \sum_m \Delta\vec{x}_e^{Bm} / M. \tag{15}$$

Increasing in the number of variant of FBG also increases with the desired matching accuracy for finding the fiducial points in a new face. But in general, the models in the FBG should be as different as possible to reduce redundancy and maximize variability. In the normalization stage we use 30 models of FBGs and the final graph extraction stage we use 70 models. These sizes seemed to give sufficient for matching accuracy and reliability.

C. Generating Face Representations by Elastic Bunch Graph Matching

After description for the individual faces and FBG, we will discuss how these graphs are generated. One of the simplest method is to select fiducial points manually. This is used for the generation of initial graphs for the system. FBG can then be formed, new images can be generated automatically by Elastic Bunch Graph Matching.

C.1. Manual Definition of Graphs

Manual definition of graphs involve three steps:

1. Mark a set of fiducial points for a given image.

2. The edges are drawn between fiducial points and edge labels are automatically computed as the differences between node positions.
3. Use Gabor wavelet transform to compute the jets for the nodes.

The set of fiducial points should cover the face evenly. For face finding, we need to use more nodes on the outline of the face. For face recognition, we place more nodes in the center of the faces because the central features are important, see Figure 9. Furthermore, we need to have optimal number of nodes with a compromise between recognition performance and speed.

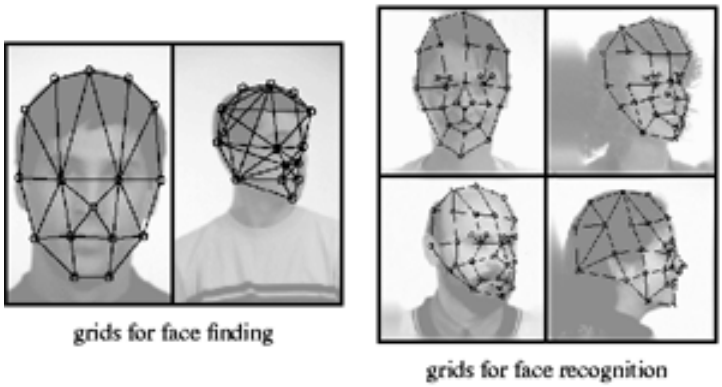

Figure 9 Object-adapted grids for different poses.

C.2. The Graph Similarity Function

The graph similarity between an image graph and the FBG of identical pose play a key role in Elastic Bunch Graph Matching. If depends on the jet similarity and the distortion of the image grid relative to the FBG grid. The similarity function is

$$S_B\left(G^I, B\right) = \frac{1}{N}\sum_n \max_m\left(S_\phi\left(J_n^I, J_n^{Bm}\right)\right) - \frac{\lambda}{E}\sum_e \frac{\left(\Delta\vec{x}_e^I - \Delta\vec{x}_e^B\right)^2}{\left(\Delta\vec{x}_e^B\right)^2} \tag{16}$$

where G^I is an image graph with node $n = 1,..., N$ and edges $e = 1,..., E$, FBG B with model graphs $m = 1,..., M$, and λ determine the relative importance of jets and metric structure. J^n are the jets at node n and $\Delta\vec{x}_e$ are the distance vectors used as labels at edges e. The first term is feature (Jet) comparison term and the second term is metric comparison term (Distortions).

C.3. Matching Procedure

The matching procedure is based on the maximization of the similarity with the FBG in Eq. (16). It needs to use a heuristic algorithm to find the optimum within a reasonable time. Here is the algorithm proposed:

1. Find approximate face position. Calculate the average magnitudes of the jets in each bunch of the FBG. Ignore the second term of the similarity function and evaluate its similarity at each location of a square lattice with a spacing of 4 pixels. The similarity function S_a without phase is used instead of S_ϕ. Re-scanning around the best fitting position with a spacing of one pixel.
2. Refine position and size. Now FBG is used without averaging. Check the four different positions with the combination of $(\pm 3, \pm 3)$ pixels displaced from the position found in Step 1, and each position check two different scales which use the same center position, a factor of 1.18 smaller or larger than the FBG average size, keeping $\lambda = \infty$. For each of these eight variations, the best fitting jet for each node is selected and its displacement according to Eq. (15) is computed. Using focus 1 only for the refinement, and the grids are rescaled and repositioned to minimize the square sum over the displacement. Find the best of the eight variations.
3. Refine size and find aspect ratio. Now the x- and y- dimensions can move independently with using the relaxation process as described for Step 2. Repeat this step by using focus 1 to focus 5.
4. Local distortion. After the above three steps, the position of each individual image node is varied to further increase the similarity to the FBG. Thus, metric similarity is used, setting $\lambda = 2$ and using the vectors $\vec{x}_e^{\,\mathcal{B}}$ as obtained in Step 3. This step only those positions, which the estimated displacement vector is small (d < 1, see Eq. (15), are interested in. For this local distortion the focus again increases from 1 to 5.

The resulting graph is called the image graph and is stored as a representation of the individual face of the image. Figure 10 gives a rough idea for the matching procedure by visually.

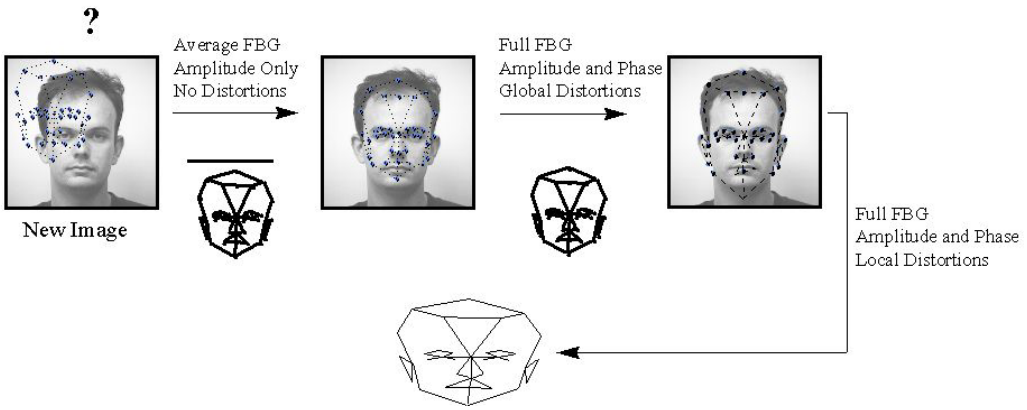

Figure 10 Matching procedures

D. Recognition

After image graph is extracted from the new images, the final process can be done. By comparing the image graph to all model graphs and picking the one with the highest similarity value. The graph similarity function is quite simple, as:

$$S_G\left(G^I, G^M\right) = \frac{1}{N'} \sum_{n'} S_a\left(J^I_{n'}, J^M_{n_{n'}}\right), \tag{17}$$

where G^I is an image graph, G^M is the model graph, node $n_{n'}$ in the model graph corresponds to node n' in the image graph. The ranking of the model graphs relative to the image graph means that how similar between them. If it is ranking one, the model will have the highest graph similarity.

IV. Experimental Result

We have using MATLAB for the implementation. As MATLAB had many build-in functions that we don't need to take care about it. These speed up the implementation time. The eigenfaces and elastic bunch graph matching were tested in recognition experiments. Here, the results will be presented and discussed, starting with the face database we used.

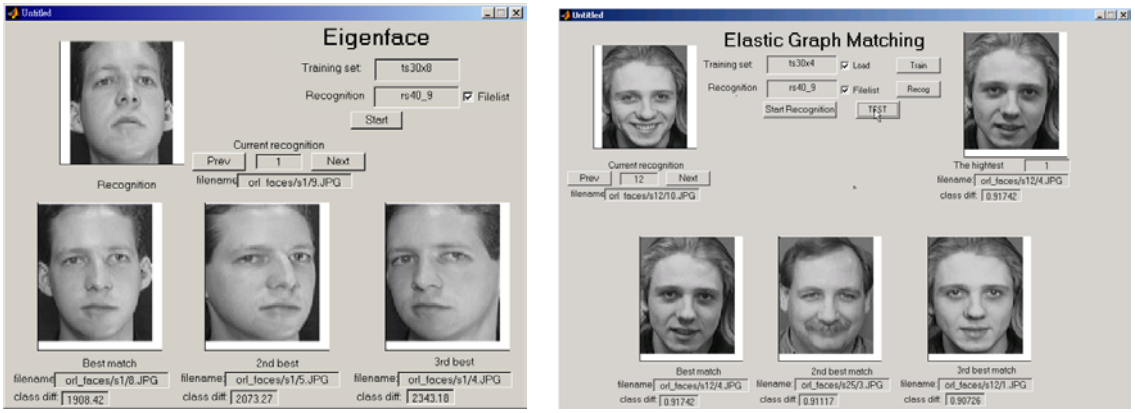

Figure 11 Snapshot of the implementation of eigenface (left) and elastic bunch graph matching (right)

A. Face Database

Moderate subject size of database, Olivetti Research Lab, was used. There are 40 subjects, each with 10 variations to form a set of 400 images. All the images have very similar size, contained frontal-view and half-profiles view.

B. Experimental Results

In the experiment, the images were divided into training set and recognition set. For the training set, the file list ts10x4.txt means that we are using the first ten subjects of the database and each subject with the first four images. For the recognition set, the file list rs40_9.txt means that we are using the ninth image of each of the subject. By using a series of training set and recognition set, and different combination of the result is shown in Table 2 for eigenface and Table 3 for elastic bunch graph matching.

By setting eigenface using 15 eigen vectors and elastic bunch graph matching using 12 nodes. The selected fiducial points (see Figure 12) are:

- Eye bows (1, 2, 4, 5)
- Pupil of eyes (3, 6)

- Nose (7, 8, 9)
- Mouth (10, 11, 12)

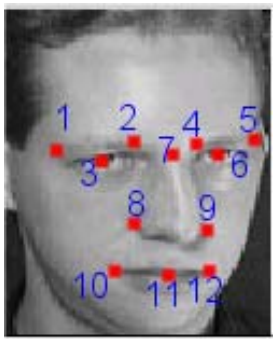

Figure 12 Example of selected fiducial point

Comparison within Eigenface

	rs40_9.txt			rs40_10.txt			Average		
	Best	2nd best	3rd best	Best	2nd best	3rd best	Best	2nd best	3rd best
ts10x4.txt	100	100	100	90	70	70	95	85	85
ts10x6.txt	100	100	100	90	100	70	95	100	85
ts10x8.txt	100	100	100	90	90	90	95	95	95
ts20x4.txt	80	70	55	85	75	65	82.5	72.5	60
ts20x6.txt	95	80	65	90	85	80	92.5	82.5	72.5
ts20x8.txt	95	95	90	90	95	80	92.5	95	85
ts30x4.txt	83.3	70	60	86.7	73.3	60	85	71.65	60
ts30x6.txt	96.7	86.7	66.7	93.3	86.7	83.3	95	86.7	75
ts30x8.txt	96.7	96.7	93.3	93.3	93.3	86.7	95	95	90
						Mean:	91.94	87.04	78.61

Table 2 Classification results for eigenface using 15 eigen vectors

Comparison within Elastic Bunch Graph Matching

	rs40_9.txt			rs40_10.txt			Average		
	Best	2nd best	3rd best	Best	2nd best	3rd best	Best	2nd best	3rd best
ts10x4.txt	90	90	50	70	60	40	80	75	45
ts10x6.txt	80	90	60	70	70	60	75	80	60
ts10x8.txt	90	80	70	70	70	50	80	75	60
ts20x4.txt	85	70	60	75	60	50	80	65	55
ts20x6.txt	85	85	65	80	80	65	82.5	82.5	65
ts20x8.txt	85	95	75	80	80	65	82.5	87.5	70
ts30x4.txt	80	60	53.3	83.3	60	50	81.65	60	51.65
ts30x6.txt	83.3	80	63.3	86.7	83.3	50	85	81.65	56.65
ts30x8.txt	80	93.3	76.7	83.3	86.7	73.3	81.65	90	75
						Mean:	80.92	77.41	59.81

Table 3 Classification results for elastic bunch graph matching using 12 nodes vectors

For the ORL database, the eigenface have a better performance (91.94% recognition), rather than that of elastic bunch graph matching (80.92% recognition). The result is not similar to that of the result from Jun Zhang [3]. Here are some possible problems in our implementation of elastic bunch graph matching. First of all, the FBG formed doesn't have enough variation in the appearance of face. As the paper proposes to include about 70 different variations include different age, gender and appearance. However, we have only 40 different subjects in the database, this may have an adverse effect in the process. Secondly, the pose of the FBG is not the same pose, this may cause the location of fiducial point have a variation. Lastly, the fiducial point we selected is only 12, which is relatively small when comparing to the original approach.

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.

V. Summary

In this paper, we have given the idea of two recently approach in face recognition, eigenface and elastic bunch graph matching. The performance of these two approaches has been illustrated by the experiment using the ORL face database. Although the result is not the same as the expectation, it still can indicate the characteristic of each approach. For the eigenface, it is a minimum distance classifier, which work well when lighting variation is small and the pose is similar. The performance deteriorates significantly as lighting variation increases, the measure of face difference is not reliable. 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.

VI. Future Work

For the face recognition, we had studied and implemented two recently approach. However, the implementation of elastic bunch graph still has a lot of problem, which we need to tackle. The problems include, getting the huge size of face database, modification on the matching procedure, make use of phase similarity function and make the computationally efficient.

Appendix A. Reference

[1] M. Turk and A. Pentland, “Eigenfaces for Recognition”, Journal of Cognitive Neuroscience, March 1991.

[2] L. Wiskott, J. Fellous, N. Kruger and C. von der Malsburg, “Face Recognition by Elastic Graph Matching”, CSC Press, ISBN 0-8493-2055-0, Chapter 11, pp. 355-396 (1999).

[3] J. Zhang, Y. Yan and M. Lades, “Face Recognition: Eigenface, Elastic Matching, and Neural Nets”, Proceedings of the IEEE, Vol. 85, No.9, Sept. 1997

[4] L. Sirovich and M. Kirby, “Low-dimensional procedure for the characterization of human faces,” J.Opt. Soc. Am. A, Vol.4, No.3, March 1987, 519-524

[5] J. Daugman, “Comple discrete 2-D Gabor transform by neural networks for image analysis and compression,” IEEE Trans. on Acoustics, Speech and Signal Processing, 36(7):1169-1179, 1988.

[6] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Von der Malsburg, R.Wurtz and W. Konen,”Distortion invariant object recognition in the dynamic link architecture,” IEEE Trans. on Computers, 42(3):300-311, 1993.

[7] D. Fleet and A. Jepson, “Computation of component image velocity from local phase information,” Int’l J. of Computer Vision, 5(1):77-104, 1990.

[8] W. Theimer abd H. Mallot,”Phase-based binocular vergence control and depth reconstruction using active vision,” CVGIP: Image Understanding, 60(3):343-358,1994.