FACE RECOGNITION USING ELASTIC BUNCH GRAPH MATCHING

Sandeep R¹, D Jayakumar²
Dept. of ECE, Kuppam Engineering College, Chittoor, Andhra Pradesh.

Abstract: Traditional automatic face recognition methods focus on handling frontal face images. They cannot be directly applied to the pose-varied or non-frontal face images captured by non-intrusive video surveillance systems. The project presents a non-frontal face recognition algorithm based on Elastic Bunch Graph Matching (EBGM). The proposed method measures face similarity using facial features which are more robust to pose variation. Experimental results show that the proposed method can achieve a verification accuracy of 97 percent on face images with 30 degree pan-angle. The proposed method can reasonably tolerate 10 percent variation in the pan-angle, indicating its robustness in tolerating errors in pose estimation. The proposed method uses three main phases modeling phase, training phase and recognition phase. We have used 15 landmarks to extract feature for face recognition. One of the most interesting results is that accuracy of the landmark localization process seems to have very little effect on the final algorithm performance. The Gabor wavelet set used in the algorithm has the most effect on algorithm performance. By the overall comparison of the results with the results of the other face recognition technique it’s found that EBGM produces the more accurate recognized output.

Keywords: Image Processing, Face Recognition, EBGM, Gabor Wavelet

I. INTRODUCTION
A face recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. An image may be defined as a two dimensional function \( f(X, Y) \) where \( X \) and \( Y \) are spatial co-ordinates. The amplitude of \( f \) at any pair of co-ordinates \( (x, y) \) is called the intensity or gray level of the image at that point \( x, y \) and the amplitude values of \( f \) are finite and discrete quantities. EBGM algorithm contains three main phases: modeling phase, training phase and recognition phase. But before these main phases, pre-processing is performed on whole database to equalize facial images. First phase of EBGM is creation of bunch graph from a modeling set using localizing landmarks manually and computing jets in each landmark location. After face modeling, system is trained by some training samples. In training phase, only one sample per class is used. The algorithm recognizes novel faces by first localizing a set of landmark features on face using bunch graph model and then measuring similarity between input features and all training samples to find the most similar face. The landmark location is refined by extracting a novel jet from the estimated location of the landmark in the novel image. The most similar jet is selected from the bunch graph and this jet then serves as a model. Both the novel jet and model jet contain frequency information about the local image region around their extraction point. Using phase information stored in the two jets, it is possible to calculate a displacement of the novel jet from the true location of the landmark. This is accomplished by finding a displacement that would make the phase information of the novel jet very similar to the phase information of the model jet. Once the landmarks have been located, a structure called a face graph is created where each node corresponds to a landmark. The landmarks are characterized by two things, a location in the image and a Gabor jet extracted from that location. After the face graph is created, the image is discarded, and the face graph becomes the internal representation of that image. The face graph occupies less memory than the image, and computing the similarity of face graphs is much faster than computing the similarity of images. For this reason, an entire database of faces can be kept in memory, and new images can be identified rapidly. The bunch graph contains Gabor jets from a set of representative images, the model images. A face graph must be created for each image to be recognized. It is the face graphs that are compared during recognition. After the algorithm has created a face graph for two images, their similarity can be computed. The base line algorithm locates and extracts local facial features, such as eyes, noses and mouth corners, for face matching. The geometric locations of the facial features are represented as a face graph Meanwhile Gabor responses of the extracted facial features are stored in the wavelet form called Gabor jets. To locate facial features, EBGM first initializes feature locations using the model bunch graph which is trained. Local regions around the refined feature locations are then extracted using multi-scale and multi-directional Gabor wavelets. In general Gabor wavelets are chosen because their responses are similar to human visual perception. Finally, EBGM calculates the similarity score between two face images by comparing their EBGM templates. It has been shown that the authentication performance of EBGM depends significantly on how accurate facial features are located.

II. LITERATURE SURVEY
The following literature surveyed papers helped us to combine the relevant equations and methods to implement "EBGM" algorithm for pose tolerant non frontal images to perform face recognition using MATLAB. Changbo Hu, Josh Harguess and J. K. Aggarwal, 2009 proposed "Patch based face recognition "which helped to
carry out face recognition from video which is a novel method for video-based face recognition by collecting face patches from video and stitch them to reconstruct a still face image. Computer & Vision Research Center / Department of ECE the University of Texas at Austin. Ming – yuan shieh, choung Ming Hsieh. 2010 proposed “PCA and LDA based fuzzy face recognition system” the paper proposed the integration of PCA and LDA for the purpose of recognition, the paper successfully accomplished a real time face detection and recognition system which could be applied for robotic system and access control system department of electrical engineering, southern Taiwan university, Taiwan. Liang yu and Yong fang, 2010 proposed “A Novel Face Representation toward Pose Invariant Face Recognition” a novel face representation approach based on the space-filling tree. The proposed structure has the better ability to represent the poses which have varied poses, School of Communication and Information Engineering Shanghai University Shanghai, China. David Monzo, Alberto Albiol, Antonio Albiol, Jose M. Mossi, 2010 proposed “A Comparative Study of facial landmark localization methods for Face Recognition using HOG descriptors” compares several approaches to extract facial landmarks and studies their influence on face recognition problems. The results obtained show that better recognition results are obtained when landmarks are related to real facial fiducial points (AAM, EBGM vs. Square grid). Detection of landmarks using AAM is better than using EBGM (see results for FERET with dup sets). ITEAM-Universidad Politecnica Valencia.

A. Motivation
Face recognition has been shown to be useful for a growing variety of tasks, notably a multi-view face recognition system which can handle pose variation, criminal identification, civilian and law enforcement applications, wearable systems like memory aids or context-aware systems etc. A non-frontal face recognition algorithm based on Elastic Bunch Graph Matching (EBGM) is proposed, this proposed method measures face similarity using facial features which are more robust to pose variation. Experimental results show that the proposed method can achieve a verification accuracy of 97% on face images with 30 degree pan-angle. Also, this method can reasonably tolerate +10 degree variations in the pan-angle, indicating its robustness in tolerating errors in pose estimation and it can be easily extended to provide non-frontal face recognition. From literature survey it’s been found that the efficiency of face recognition using EBGM technology is more efficient compared to PCA, SVM, FLDA. and LDA.

III. IMPLEMENTATION
A. Pre-processing
Nothing but normalization where the image captured is converted into a standard format using a frequency domain filtering for enhancement or compression. So the main aim of this preprocessing is resizing the input image.

Step 1: capturing raw image data

Here the input given is nothing but digital image.

Figure 1: a raw input data
The above image is an example of input image which is having variable size, background illumination and format. But this type of image is not suitable for the comparison with the stored images of the data base so we normally make use of a standard form of an input image so that the comparing of the images for recognition becomes easy, which is done by preprocessing.

Step 2: Resizing the input image
The image captured will be converted into a standard size of 92 x112 which is same as that of the size of the images stored in the trained data base

Figure 2: A Resized Image
Step 3: Cropping the input image
A resized image may contain different parts of the body so we actually select only the Head part of a person so that the extraction of the comparable features becomes easier.

Figure 3: A cropped image
Step 4: Gray scale conversion of input test image
The input test may be a color image most of the time but these color images are not suitable for comparison so they are converted into a grayscale image by making use of a Gabor filter which is the most suitable filter for the gray scale conversion of an input image and after the gray scale conversion we are going to extract the real, imaginary and the magnitude part of the input image.

Step 5: Normalizing the input image
The input image captured for the purpose of recognition may be of varied size, illumination and clarity so these variations become the major parameters which directly affect the accuracy or the performance of the recognition system. so in order to achieve accuracy and better recognition rate we actually convert the input image into a standard format using
a spatial or frequency domain filters for enhancement or compression according to the requirement. This pre-processing of the input image is also called as the Normalization. Here in the normalization procedure we are making use of a Gaussian and edge filters for converting the image into a gray scale image which is the standard form of the input image for comparing it with the images stored in the data base. By making use of Gaussian filter, ringing effect and other higher frequency components can be eliminated and by Using edge filtering, abrupt changes in gray level are associated with high frequency components can be eliminated so image sharpening can be obtained in frequency domain by a high pass filtering.

Figure 4: Dataflow diagram.

After gray scale conversion using a frequency domain filters we crop the image into the standard size of 92x112 pixels which is same as that of the size of the images stored in the trained data base and also the background illumination of the image captured will be neglected The same steps are performed for the images which are also stored in the data base and finally the sharpened image is nothing but the combination of the Gaussian and the edge filtered image thereby making it easy for the purpose of comparison and producing the perfect recognition at the output.

B. Feature extraction
All the mentioned preprocessing steps are applied for both the test and the trained images so that the images when used for feature extraction and comparing the features followed by recognition contains the same format of data which made the recognition a much simple job. The main features we are concentrating are the jets at the fiducial points of the face which are manually selected for the test input images but automatically selected for the trained images for comparing the test and the trained images for recognition. The steps involved for extracting the jet features and using it for recognition are given below.

C. Extraction of jets
Jets are selected by hand to serve as facial features for the test images.

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

(1)

With a family of Gabor kernels

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

(2)

In the shape of plane waves with wave vector $\vec{k}_j$, restricted by a Gaussian envelope function. We employ a discrete set of 5 different frequencies, index = 0…4, and 8 orientations, index $\mu = 0…7$.

$$\psi_j(\vec{x}) = \left(\begin{array}{c} k_j \cos \varphi_{\mu} \\ k_j \sin \varphi_{\mu} \end{array}\right)$$

(3)

With index $j = \mu + 8v$. This sampling evenly covers a band in frequency space. The width $\gamma/k$ of the Gaussian is controlled by the parameter $\gamma = 2\pi$. This is known as a wavelet transform because the family of kernels is self-similar, all kernels being generated from one mother wavelet by dilation and rotation. A jet is defined as the set of complex coefficients obtained for one image point. It can be written as

$$J_j = a_j \exp(i \varphi_j)$$

(4)

Gabor wavelets were chosen for their robustness as a data format and for their biological relevance since they are DC-free, they provide robustness against varying brightness in the image. Robustness against varying contrast can be obtained by normalizing the jets. The limited localization in space and frequency yields a certain amount of robustness against translation, distortion, rotation, and scaling. Only the phase changes drastically with translation.
D. Landmarks

Landmark Locations are the pixel coordinates of landmarks. Landmark locations define the geometry of the face. An example of this is the nose tip, which has a well-defined location. Landmark Features are defined by the frequency information of the local regions that surround the landmark locations. A landmark jet is not a particular point but instead contains information about the pixel values surrounding the landmark location. A landmark jet refers to information on what the landmark looks like. Gabor jets are used to represent the landmark jet information in the EBGM algorithm. A Gabor jet is produced by convolving the landmark location with a collection of Gabor masks. Therefore, the Gabor jet will contain a good description of the local frequency information around the landmark. The structure of Gabor wavelets allow this information to be heavily weighted in the area immediately surrounding the landmark, while still covering enough of the image to get a good description of the landmark. Gabor jets act as feature vectors that describe the landmark from which the jet was taken. Landmark similarity is based only on the Gabor jets taken from the two landmarks. The jets are based on a variety of Gabor masks and contain an accurate description of the landmark. Therefore, the similarity of jets is a good indicator of the similarity of the landmarks.

a). Landmark Localization

This section will present a high level discussion of how the process is implemented. Landmark localization has two parts. The first is to estimate the new location based on the locations of known landmarks in the image. The second is to refine that guess based on the phase information from a Gabor jet extracted from the estimated location. When a new image enters the system, it is first normalized using the eye coordinates. The eye coordinates are transformed to line up in the same points. These points are a very good estimate of the locations of the eyes, and for this reason, the algorithm always locates the eye landmarks first. To locate the eye landmarks the algorithm extracts a jet from each of the eye coordinates locations. It then uses the bunch graph and one of the landmark localization methods. The algorithm locates points near the eye coordinates first, because the estimates of the eyes are very reliable. The algorithm then works radially outward until it reaches the edge of the head. For example, once the eye landmarks are located, the algorithm uses those points to estimate the location of the bridge of the nose and then refines that guess using the bunch graph and Gabor jet phase information. The next point placed is the left eye brow peak, and the location estimate is based on the previous three located points. This process continues for each landmark where every new estimate is based on all of the previously located points. Once the algorithm has estimated the location of a landmark, it needs to refine that guess using a displacement estimation method. First a jet is extracted from the estimated location. The jet is compares to every corresponding jet in the bunch graph using the chosen phase based similarity. The model jet with the highest similarity is selected and is used to estimate the displacement. Finally, the displacement is used to adjust the landmark location.

Because we manually selected the points for the model images we feel that we should comment on the placement of these points. We noticed the following problems with the images in the figure:

- Right Jaw is too far right. Left Eye Brow Outside is too far left. Right Eye Brow Peak needs to be centered. Center Top Head is too low. Center Nose Tip too high.
- Left Jaw and Right Jaw misplaced. Left Eye too far left
- Right Jaw too far right. Right Nose Bottom and Right Mouth Corner too far up and right. Center Top Head is too low.

With the exception of the jaw line for image “b.” all of these mistakes seem to be minor. Many of them were difficult to spot. For each model image, jets are extracted from the manually selected landmark locations and are then added to the bunch graph. After the bunch graph is created, each novel image is loaded, and the landmark locations are found. For each image, a file is saved to disk that only contains the locations of the landmarks. Because the images are processed one at a time, the code is very memory efficient. The process can also be easily divided for parallel processing by running multiple executables on different sets of novel images.

b). Model set selection

Analyzing the selection of model images that are used to create the bunch graph. There are two factors tested in this model set configuration study. The first was the size of the model set. This first experiment used the basic algorithm using model sets of different sizes. The second tested the effect of using model images that are also in the testing set. The result is that there seems to be very little effect of the localization process on algorithm performance when using the standard jet based face graph similarity measure.

c). Bunch Graph Creation

Each node of the bunch corresponds to a facial landmark and contains a bunch of model jets extracted from the model imaginary.

To find fiducial points in new faces, we used a general representation rather than models of individual faces. This representation covers a wide range of possible variations in the appearance of faces, such as differently shaped eyes, mouths, or noses, different types of beards, variations due to sex, age, race, etc. It is obvious that it would be too expensive to cover each feature combination by a separate
graph. We instead combine a representative set of individual model graphs into a stack-like structure, called a face bunch graph (FBG). Each model has the same grid structure and the nodes refer to identical fiducial points. A set of jets referring to one fiducial point is called a bunch. An eye bunch, for instance, may include jets from closed, open, female, and male eyes, etc., to cover these local variations. During the location of fiducial points in a face not seen before, the procedure described in the next section selects the best fitting jet, called the local expert, from the bunch dedicated to each fiducial point. Thus, the full combination of jets in the bunch graph is available, covering a much larger range of facial variation than represented in the constituting model graphs themselves.

A bunch graph is created by collecting model Gabor jets for every facial landmark and storing them in a graph-like structure. The algorithm uses manually selected landmark locations to produce the bunch graph. The manually selected landmark locations are collected by a tool, with a graphical user interface, that allows a user to select points on an image that correspond to landmarks. The output of the graphical tool is a set of files that contain the landmark locations for certain images. Each file corresponds to an image that was processed manually and contains a set of pixel locations that correspond.

To create the bunch graph, the algorithm loads each image that is used as a model. It then extracts a model jet from each manually selected landmark location and adds the jet to the appropriate bunch. Each node in the bunch graph contains a group of Gabor jets corresponding to one landmark. The jet bunches can then provide jets to serve as model landmarks when locating landmarks in novel images. One powerful aspect of the bunch graph is that each bunch contains many different examples of its particular landmark jet. For example, the algorithm can provide model jets of eyes with glasses, eyes that are closed, and eyes under different lighting. Furthermore, the algorithm can select jets from different subjects for each landmark on the face. For a particular face the algorithm can mix and match eyes, noses, and mouths such that each landmark has the best possible example and, therefore, the best possible localization.

Although the selection process seems straightforward, details tend to complicate the training process. The two most prominent details are selection of model images and selection of landmark locations. For each, 15 landmark locations were selected by hand around the eyes, nose, mouth, and the edge of the head.

d). Face Graphs

A face graph is the structure used to represent a face. Every test image in the system has a corresponding face graph. The graph has one node for each landmark in the image. Every node in the graph contains the location of the landmark and a Gabor jet extracted at that point. The algorithm computes the similarity of two faces using the data stored in the face graphs. Face graphs are compared in order to compute the similarity between images. Similarity can be computed as a function of the landmark jets, landmark locations, or both. A face graph also contains jets that were extracted at locations interpolated between landmarks. For example, there are no landmark locations on the cheeks or forehead. These regions include much of the surface area of the face. One of the simplest ways to compute the similarity between two sets of landmark locations is to compute the sum of the Euclidean distances between the locations. Since the images have already been aligned using the eye coordinates of the subject, the landmark locations should therefore be aligned relatively well, and Euclidean distance can be used to measure similarity.

$$\text{Euclidean distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Where \((x_1, y_1)\) is the location of the \(i\)th landmark in the first face graph, and \((x_2, y_2)\) is the location of the landmark in the second face graph.

e). Gabor Wavelets

A Fourier transform decomposes a signal such that it can be represented as a combination of sinusoids. It is very useful in signal processing because there are many things that a frequency space analysis can reveal about a signal that are not obvious from the original data. It is important to understand the phase angle of the coefficient. Much of the EBGM algorithm relies on this polar coordinate transformation of the coefficients to estimate displacements of image points. If wavelet coefficients are computed at two points that are separated by a small displacement, the difference in the phase angle of the coefficients should be roughly proportional to the displacement between the two convolution points.
E. Recognition
Comprises of the technique used for comparing the input image with the trained images, i.e. EBGM which include.

Similarity Measurement: The final step of the EBGM algorithm is to produce a distance matrix for the database. All of the analysis techniques for the Face Recognition Evaluation System utilize the distance matrix to determine system performance.

Identification: In the system identification is based on nearest neighbor classification. The output of the algorithm is a distance matrix which specifies the similarity between each image in the database. The system assumes algorithms are structured as nearest neighbor classifiers where more similar images have smaller distances between them. The system is not intended to serve as a real time face recognition system. It is intended to analyze the performance of face recognition systems based on well-designed experiments.

-Schedule of Graph extraction
To minimize computing effort and to optimize reliability we extract a face representation in two stages, each of which uses a matching procedure. The first stage called the normalization stage has the purpose of estimating the position and size of the face in the original image so that the image can be scaled and cut to standard size. The second stage takes this image as input and extracts a precise image graph appropriate for face recognition purposes. The two differ in emphasis .the first one has to deal with greater uncertainty about size and position of the head and has to optimize the reliability with which it finds the face , but there is no need to find fiducial points with any precision or exact data important for face recognition . The second stage can start with little uncertainty about position and size of the head, but has to extract a detailed face graph with high precision. The second stage uses the matching procedure starting the match at standard size and position. The face bunch graph used in this stage have more nodes, which we placed in positions we believe are important for person identification, emphasizing the interior of the face. Each of the three principal poses (frontal, half profile, and left – facing poses are flipped to right – facing poses) is matched with a different grid

-Matching procedure
The goal of elastic Bunch Graph Matching on a Probe image is to Find the Fiducial points and thus to extract from the image a Graph Which Maximizes the Similarity with the FBG .Here we use Phase information and increase the focus of displacement estimation.

Step 1: Find approximate Face position
Condense the FBG into an average graph by taking the average magnitudes of the jets in each bunch of the FBG .we use this as a rigid model and evaluate its similarity at each location of a square lattice .repeat the scanning around the best fitting position with spacing of 1 pixel. The best fitting position finally serves as the starting point for the next step.

Step 2: Refine position and size
Now the FBG is used without averaging, varying it in position and size. we check the four different pixels displaced from the position found in step 1 ,and at each position check two different sizes which have the same center position .this is done without any effect on metric similarity .and we keep the best of different variations as the starting point for the next step.

Step 3: Refine size and find aspect ratio
A similar relaxation process as described for step 2 is applied, but relaxing the x- and y- dimensions independently. In addition, the focus is increased successively.

Step 4: Local distortion
In pseudo – random sequence the position of each individual image node is varied to further increase the similarity to the FBG .Now the metric similarity is taken into account .in this steps only those positions are considered for which the estimated displacements are small.

- The Graph Similarity function
A key role in elastic Bunch Graph Matching is played by a function evaluating the graph similarity between an image graph and the FBG of identical pose. It depends on the jet similarities and the distortion of the image grid relative to the FBG grid. For an image graph with nodes n=1 ...N and edges e = 1 …E and G with model graph m = 1 ...M the similarity is defined as

$$S_n(G,F) = \frac{1}{N} \sum_{n=1}^{N} \max(S_d(J_n^1, J_m^1), \ldots, S_d(J_n^E, J_m^E)) - \lambda \sum_e (\delta x_e \cdot \delta x_e) \cdot \sqrt{\delta y_e^2 + \delta y_e^2}$$

where λ determines the relative importance of jets and metric structure J_n are the jets at nodes n, and δx_e the distance vectors used as labels at edges e, since the FBG provides several jets for each fiducial point the best one is selected and used for comparison .these best fitting jets serve as local expert for the image face .

- Comparison of the features of trained and test images
After having extracted model graphs from the gallery images and image graphs from the probe images, recognition is possible with relatively little computational effort by comparing an image graph to all model graphs and picking one with highest similarity value. A comparison against gallery of individuals took slightly second. The similarity function we use here for comparing graphs is an average over the similarities between pair of corresponding jets. For image and model graphs referring to different to different pose, we compare jets according to manually correspondences. if G^i is the image graph, G^m is the model graph and node n in the model graph corresponds to node n’ in the image graph, we define graph similarity as.

$$S_G(G^i,G^m) = (1/N) \sum_{n=1}^{N} S_d(J_n^i, J_n^m)$$

Where the sum runs only over N’ nodes in the image graph with a corresponding node in the model graph. We use the jets similarity function without phase here. It turned out to be more discriminative, possibly because it is more robust with respect to facial expression and other variations.
Here we ignore the jet distortions created by rotation in depth; the graph similarity induces a ranking of the model graphs relative to an image graph. The person is recognized correctly with the correct model yields the highest graph similarity.

![Figure 10: final recognition result.](image1)

IV. CONCLUSIONS

This face recognition method describes the implementation of the Elastic Bunch Graph Matching algorithm used in face recognition process. The performance of ebgm algorithm is better compared to all other algorithms. The selective weighting strategy was made to measure face similarity using features which are more robust to pose variation. The selective weighting strategy was compared with uniform weighting strategy for non-frontal face verification. The experimental result shows that this simple weighting approach can allow the proposed non-frontal EBGM to handle face images with ±10 degrees of toleration. We have used 15 landmarks to extract feature for face recognition. One of the most interesting results is that accuracy of the landmark localization process seems to have very little effect on the final algorithm performance. The EBGM algorithm landmark localization process finds the landmarks; however, it is not very precise. The Gabor wavelet set used in the algorithm has the most effect on algorithm performance. So by the overall comparison of the results with the results of the other face recognition technique it’s found that EBGM produces the more accurate recognized output.

REFERENCES


