IJCAES
ISSN: 2231-4946
Volume III, Special Issue, August 2013

International Journal of Computer Applications in Engineering Sciences
Special Issue on National Conference on Information and Communication (NCIC'13)
www.caesjournals.org

Component Analysis Based Facial Expression Recognition

G.Sethuram Rao[1], M.Jeeva Bharathi[2], Sanjay Kumar Suman[3]

Abstract: The Intelligence of Human-Computer Interaction is one of the hot researching areas. Facial expression recognition is an important part of human-computer interaction. At present, the research of facial expression recognition has entered an era of a new climax. In real-time facial expression recognition system, the paper presents an expression feature extraction method that combined canny operator edge detection with the AAM(active appearance model) algorithm. During the Canny edge detection, the adaptively generated high and low thresholds, increased the capability of noise suppression, and the time complexity of the algorithm is less than the traditional canny operator. Finally, by using leas squares method, we can classify and identify the feature information

I. Introduction

Besides fingerprints and iris, faces are currently the most important and most popular biometric characteristics observed to recognize individuals in a broad range of applications such as border control, access control and surveillance scenarios A facial 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. One of the ways to do this is by comparing selected facial features from the image and a facial database. It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems.[The method is combining the AAM and adaptive Canny operator edge detection to carry out the expression feature extraction. When using the Canny operator to detect the edge of the facial image, the algorithm can adaptively generate the dynamic threshold according to the edge gradient information of sub-image and combined the feature information of global edge gradient. So it needn't human intervention.

II. Related works

In recent years face recognition has received substantial attention from both research communities and the market, but still remained very challenging in real applications. A lot of face recognition algorithms, along with their modifications, have been developed during the past decades. While this growth largely is driven by growing application demands, such as identification for law enforcement and authentication for banking and security system access, advances in signal analysis techniques, such as wavelets and neural networks. At present many of the face recognition methods have good results on the static image recognition [1, 2]. Recognition methods for the image sequence have also emerged [3]. Another biologically inspired algorithm that is commonly used for face recognition is the genetic algorithm (GA). While a neural network mimic the function of a neuron, genetic algorithms mimic the function of chromosomes. Like neural networks, genetic algorithms are only suited for the recognition of a limited number of individuals and are generally not too scalable.

III. Face Recognition Process

"The variations between images of the same face due to the illumination and viewing direction are almost always larger than image variations due to change in face identity". Face recognition scenarios can be classified into two types:

(i) Face verification (or authentication) and
(ii) Face identification (or recognition).

Enrollment process:

Enrollment process is the process of collecting the faces of persons. For this purpose, the system uses sensors or cameras to capture the images (faces), which were placed at railway stations, airports, police stations and passport offices etc. From the captured face images, the templates are generated by processing it. These templates (edge images) are stored in a database. The main aim of enrollment process is to collect images .The collection of images is limited by the particular application. The captured image features are extracted and compared with the available database. This process is shown in figure1.

First, a user must be enrolled in the system so that his biometric template can be captured. This template is securely stored in a central database or a smart card issued to the user.

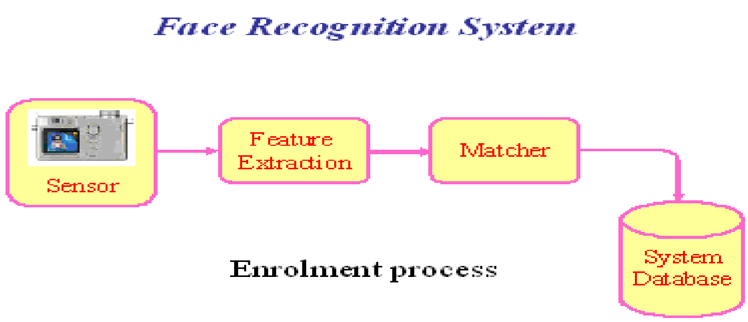


Figure 1: Enrollment process

The template is retrieved when an individual needs to be identified. Depending on the context, a biometric system can operate either in verification (authentication) or an identification mode.

Face identification:

In an identification application, the biometric device reads a sample and compares that sample against every record or template in the database. This type of comparison is called a "one-to-many" search (1: N). This is illustrated in figure2

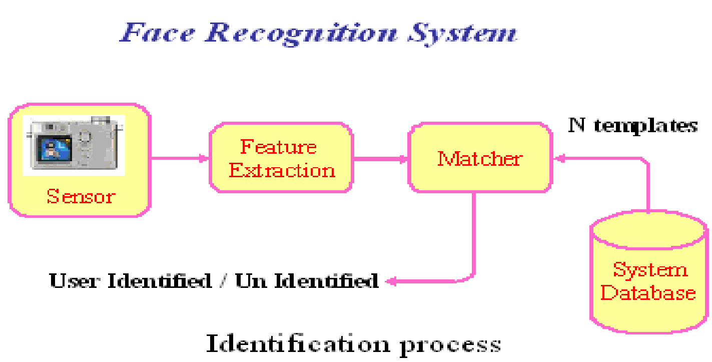


Figure 2: Identification process

Face verification:

Face verification is a one to one match that compares a query face image against a template face image whose identity is claimed. Verification occurs when the biometric system asks and attempts to answer the question "Is this X?"

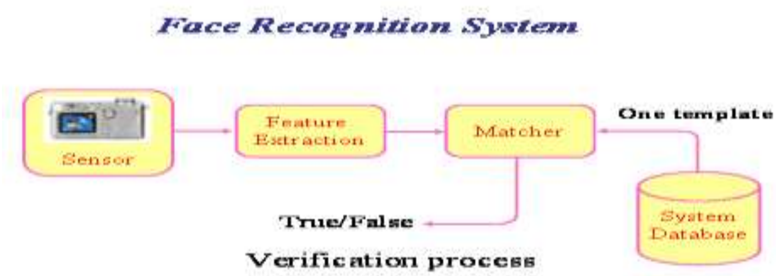


Figure 3: Verification process

After the user claims to be X. In a verification application, the biometric system requires input from the user, at which time the user claims his identity via a password, token, or user name (or any combination of the three). The system also requires a biometric sample from the user. It then compares the sample to or against the user-defined template. This is illustrated in figure 3.

IV. Principal components analysis method

Principle Components Analysis:

PCA is an algorithm that treats face recognition as a two dimensional recognition problem. The correctness of this algorithm relies on the fact that the faces are uniform in posture and illumination. PCA can handle minor variations in these two factors, but performance is maximized if such variations are limited. The algorithm basically involves papering a face onto a face space, which captures the maximum variation among faces in a mathematical form. During the training phase, each face image is represented as a column vector, with each entry corresponding to an image pixel. These image vectors are then normalized with respect to the average face. Next, the algorithm finds the eigenvectors of the covariance matrix of normalized faces by using a speedup technique that reduces the number of multiplications to be performed. This eigenvector matrix is then multiplied by each of the face vectors to obtain their corresponding face space paperions. Lastly, the recognition threshold is computed by using the maximum distance between any two face paperions. In the recognition phase, a subject face is normalized with respect to the average face and then papered onto face space using the eigenvector matrix. Next, the Euclidean distance is computed between this paperion and all known paperions. The minimum value of these comparisons is selected and compared with the threshold calculated during the training phase. Based on this, if the value is greater than the threshold, the face is new. Otherwise, it is a known face.

Principal component analysis technique can be used to simplify a huge data set consisting of a number of correlated data values into a smaller set of uncorrelated values by reducing the dimension of the data set while still retaining most of the inherent data. The process begins with a collection of 2-D facial images into a database called the face space. Using the Eigen feature method, the similar and dissimilar variations of the faces are obtained as a mean image of all the faces in the face space. This gives the training set of images. The difference image is computed by subtracting the mean image from every image in the training set, which provides the covariance matrix. From the covariance matrix the eigenvectors and eigenvalues are obtained. The eigenvectors are treated as eigen faces, which are papered into the eigen space. Then weight vectors for all the eigen faces in the eigen space are calculated. The weight vector for the test image is also calculated and the distance between them determines the presence or absence of a similar face in the face space.

Face recognition is a primary element of the many aspects of a society. If the population is limited as in small villages, identifying a particular person would not be a problem. But as the number of individual's increases, recognizing a specific face out of many becomes much difficult task. Giving a unique electronic identity to each one of them overcomes the problem to some extent but even they cannot work if the cards are stolen or passwords are forgotten. Instead, if physical aspects like facial features are taken as the basis for recognition, it would be easier to identify the faces. As much, face recognition techniques have gained a wide importance with the increase in the number of applications like security access control, criminal identification, and human–computer interaction and surveillance systems.

4.1 Mathematical Modeling

Let f (x, y) be a two-dimensional function, where x and y are spatial coordinates and the amplitude of f at any pair of coordinates (x, y) is called the intensity of the image at that point. Every image I can be expressed as a matrix of f (x, y)s of dimension m X n. Let F be the set of M training images of same dimension expressed as a array of dimension (m X n) X M

F = (I1 I2 I3………….I M) (1)

These M images are converted as vectors X i, 1 □ i □ M of dimension N (= m X n) where X i is an N X 1 vector corresponding to the image Ii in the image space F. Now F becomes

F = (X1 X2 ………..XM) (2)

The mean image □ is calculated by summing all the training images and dividing by the number of images with dimension (N X 1) as follows

$$\overline{X} = \frac{1}{M} \sum_{i=1}^{M} X_i \qquad (3)$$

The difference image is obtained by subtracting the mean image from all the training images X i is stored in a vector □ i

$$\Phi_i = X_i - \overline{X} \qquad (4)$$

The main idea behind the eigenface technique is to exploit the similarities between the different images. For this purpose the covariance matrix Cv with dimension N X N is defined as

$$C\,v = \frac{1}{M}\sum_{i=1}^{M} \Phi_i . \Phi_i^T = \quad T \quad ^{T} \quad T \qquad (5)$$

Where T = (□ 1 □ 2 □ 3 ...□ M) is of dimension N X M. This covariance matrix dimension will normally be huge matrix, and full eigenvector calculation is impractical

Figure 4: Training set of face images

4.2 Dimensionality Reduction in Covariance Matrix

We assume that without loss of generality of the whole training set, we can reduce the dimensionality of the covariance matrix B with dimension M X M defined as

$$B = T^{T}\, T = \frac{1}{M}\sum_{i=1}^{M} \Phi_i . \Phi_i^{T} \qquad (6)$$

The eigenvectors of the covariance matrix C v are computed by using the matrix B. Then the eigenvector Y i and the eigenvalue $\square^{i}$ of B are obtained by solving the characteristic eigenvalue problem | B - $\square$ I | = 0

$$B.\, Y_i = \square^{i}\, .\, Y_i \qquad (7)$$

Substituting the value of B in eqn. (6.7)

$$T^{T}\, T.\, Y_i = \square^{i}\, .\, Y_i \qquad (8)$$

Multiplying both sides with T in eqn. (8), we obtain

$$T.\, T^{T}\, T.\, Y_i = T.\, \square^{i}\, .\, Y_i \qquad (9)$$

Since $\square^{i}$ is a scalar quantity,

$$T\, T^{T}\, .\, T\, Y_i = \square^{i}\, .\, T\, Y_i \qquad (10)$$

Substituting the value of $Cv = T\, T^{T}$ in eqn.(10), we obtain

$$C_v .\, T\, Y_i = \square^{i}\, .\, T\, Y_i \qquad (11)$$

Now, let □ ^{i} = T Y i and □ ^{i} are M eigenvectors and eigenvalues of C v. In practice, a smaller set of M' ((M – 1) < M) eigenfaces is sufficient for face identification because the subsp ace is the basis for the facespace, i.e., we can represent the original face as a linear combination of these M' vectors. Hence, only M' significant eigenvectors of Cv, corresponding to the largest M' eigenvalues are selected for the eigenface computation thus resulting in a further data compression. The remaining (N – M') eigenvectors would have associated eigenvalues very close or equal to zero.

4.3 Recognition

Now, the training images are papered into the eigen face space and the weight of each eigenvector to represent the image in the eigen face space is calculated. The weight is simply the dot product of each image with each of the eigenvectors.

Wk = □ $^{T}_{k}$. □ i = □ $^{T}_{k}$. (Xi - $\overline{X}$) (12)

Where k = 1, 2 …. M' . All the weights are converted in the form of a matrix with dimension M' X 1

□ = [w1, w2, w 3………..wk]^{T} (13)

To test an unknown image □ , we evaluate its weight (wi) by multiplying the eigenvector (□ i) of the covariance matrix (C v) with difference image (□ – $\overline{X}$)

W i = □ $^{T}_{i}$ (□ – $\overline{X}$) (14)

Now the weight matrix of the unknown image becomes

□ = [w1, w2, w 3………..wM']^{T} (15)

The Euclidean distance □ k between unknown image and each face class is defined by

□ k2 = || □- □ k || 2; k = 1,2,…….,Nc (16)

Where Nc is the number of face classes. We obtain a reconstructed image by multiplying the weight matrix (□) of the unknown image with the eigenvector matrix (□) of the covariance matrix (Cv) and adding the mean face image

($\overline{X}$) to it. □ f = □ .□ +$\overline{X}$ (17)

Where □ = [□ 1 □ 2 ……… □ M']. Now the Euclidean distance (□) between the original unknown image and the reconstructed image is computed in order to distinguish between face images and non-face like images.

□ 2 = || □– □ f || 2; (18)

Calculating the Euclidean distance between two data points involves computing the square root of the sum of the squares of the differences between corresponding values. These Euclidean distances are compared with an empirical threshold value. If the threshold value is greater than both the Euclidean distances, the system recognizes it as a face belonging to a known class. If the threshold value is less than the Euclidean distance (□ k) and greater than □ , then it recognizes the face as close as to the facespace but do not represent a known face class. If the threshold value is less than both the Euclidean distances, the system recognizes it as unknown object

5 .Algorithm of PCA method

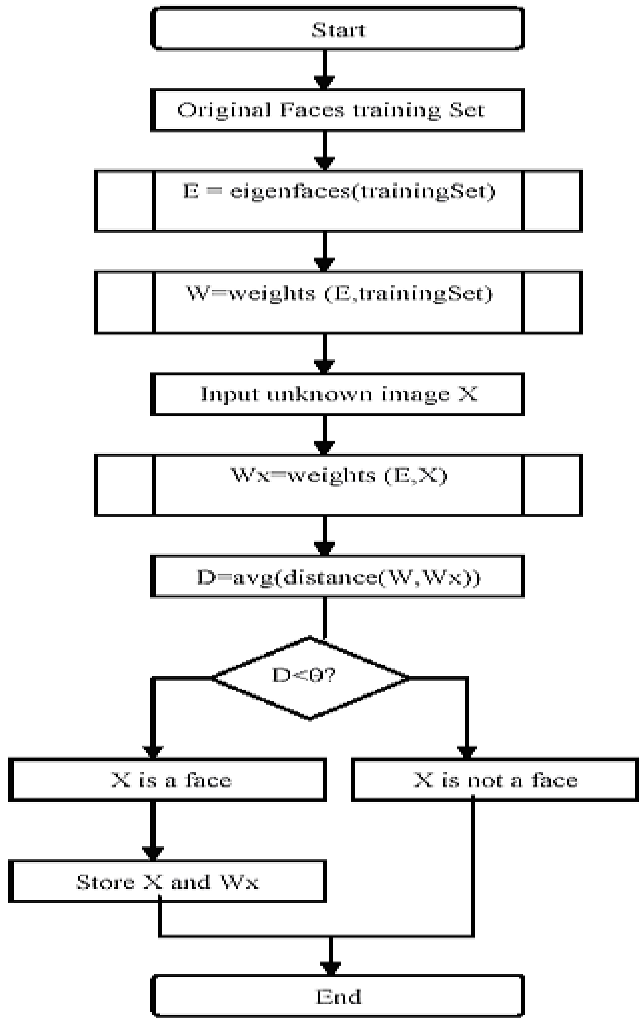


Figure 5 Algorithm of PCA method

6. RESULTS AND DISCUSSIONS

The database used in this context contains a set of face images of 130 individuals each with 3 different expressions and poses under similar illumination. The dimension of each image is 200 X 150 pixels in gray mode. PCA technique is applied to the database with a test set of random images. Eigen faces represent prominent features of face images of the training set. This plot corresponds to highest eigenvalues of the individual Eigen faces of the training set. The highest eigenvalues representing the Eigen faces of 80 individuals with a training set of one face per individual is plotted in figure 7. Face database is used for the computation of the above plot. From the figure 7, it is observed that 20% of the faces are sufficient for recognition and the remaining 80% faces are insignificant. Even if they are discarded, there is no problem for recognition. In order to find out the proper value of M ', the eigen values of the covariance matrix is ranked in descending order. Suppose there are M ' eigenvalues, i.e., □ 1, □ 2, □ 3, …□ M', (it is convenient if these are appropriately arranged such that □ 1 □ □ 2□ □ 3□ □.M').where □ 1 is the largest eigen value while □ M' is the smallest one. The eigen value range of the training data provides useful information for the choice of the reduced PCA space M'. It is also observed from Figure 7, that the Eigen value variation of the training face images increases gradually as the face space grows from 1 to 80 with 10. In the figure 7, the number of Eigen faces taken the x-axis and take the Eigen value variation on the y-axis. The graph shows that how the eigenvalues changing from one face image to the other face image. As the number of the Eigen faces in the database increases, the Eigen value is observed to have exponential decay till it reaches the value zero. For the first Eigen face, the eigen value is the highest and for the second eigen face, it is the next highest and so on and for the last eigen face, it is zero. This means that the eigen value variation is inversely proportional to the number of eigen faces with the Eq.(5.11) is presented in figure 7.

When the Euclidean distance increases, the recognition rate is found to be decreasing. That is, for the minimum distance, the recognition rate is the highest. From the figure 7, it is observed that as the distance goes on increasing, the recognition rate drops correspondingly until it reaches zero, which means that the recognition rate is inversely proportional to the Euclidean distance with the Eq.5.18 is presented in figure 7.When experimented with the number of eigenvectors and recognition rate, results show that the recognition rate grows as a factor of individual number of eigen faces. For a single set of individual Eigen faces, the recognition rates is lower but as the sets are added, the recognition rate increases correspondingly with it. From the figure 8, it is observed that the recognition rate is directly proportional to the individual Eigen faces.

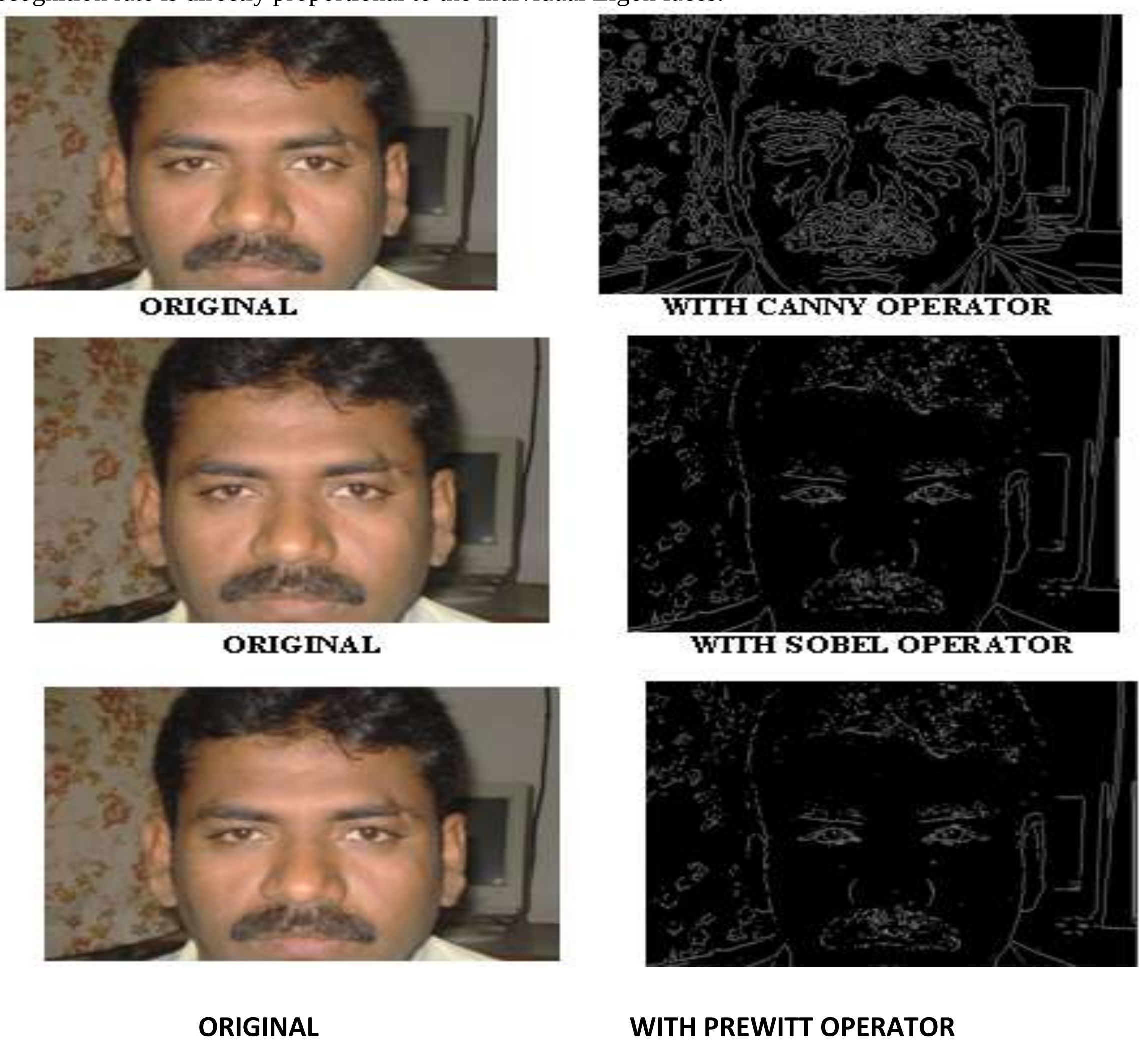


Figure 6 Edge images with different operators

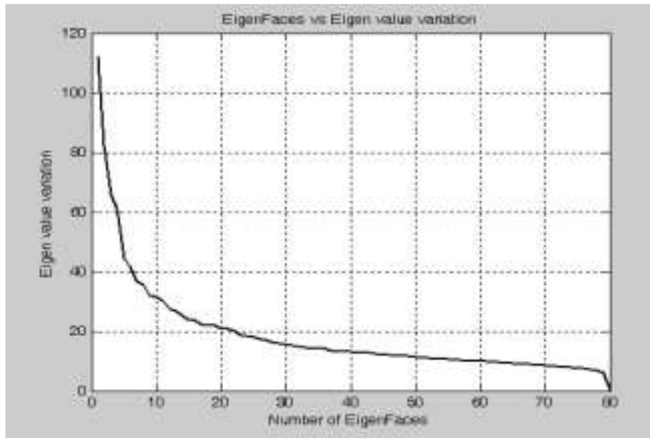


Figure 7: Eigen faces VS Eigen value variation

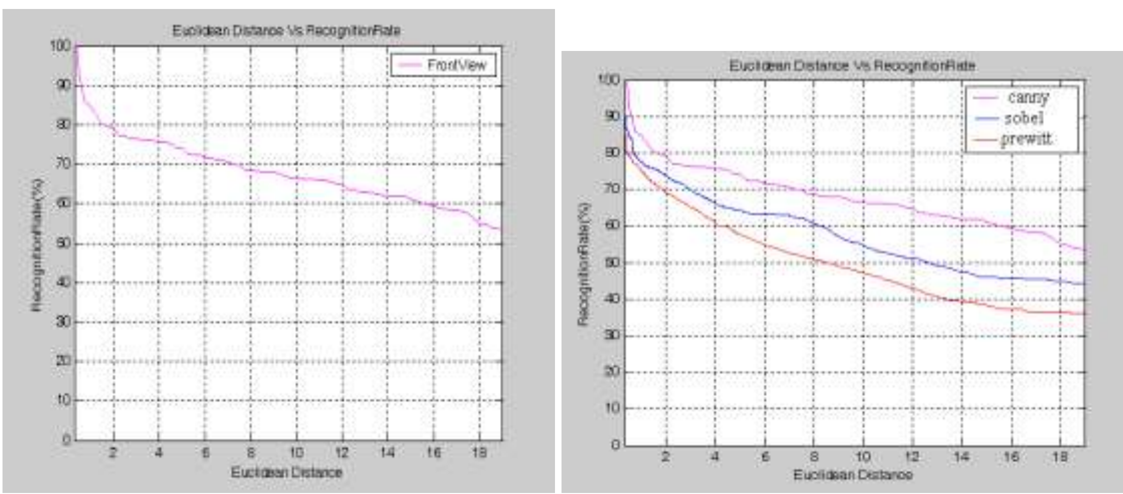


Figure 7: Euclidean distance VS Recognition rate

For our experiments we have considered 200 images from the entire face database for which the timing diagram is illustrated in the figure 7 The figure shows timing plots for both the cases – with covariance matrix reduction and without covariance matrix reduction. It is observed from the graph that the timing for face recognition with covariance matrix reduction is around 34 seconds while without eduction of the covariance matrix, it is around 176 seconds.

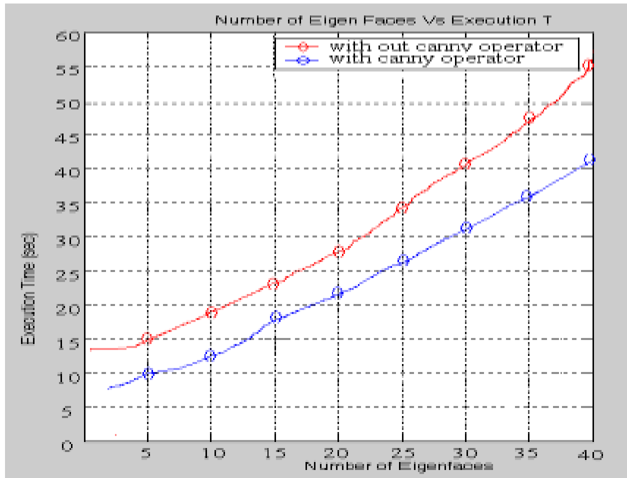
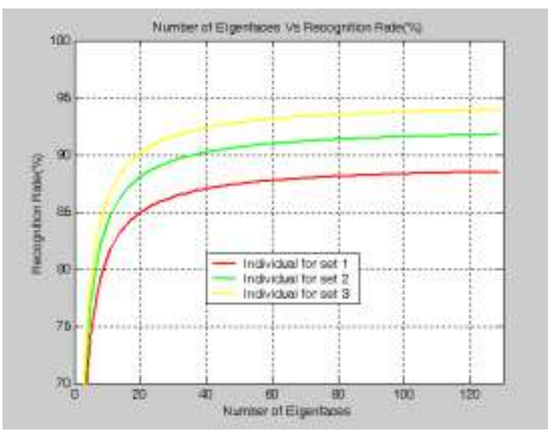


Figure 8: comparison of with and without canny operator



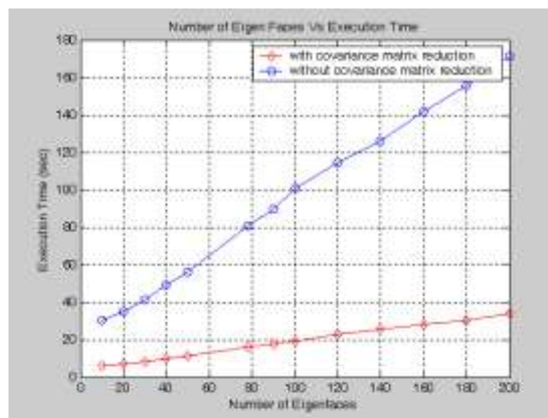


Figure9: Number of Eigen faces Figure 10: Execution time with and without

VS covariance matrix dimension reduction Recognition rate

This shows that without covariance matrix reduction, the execution time for the face recognition algorithm is almost five times that of with covariance matrix reduction.

6 Conclusion and future work

This paper gives an elaborate discussion about modeling and results of face recognition algorithm using principal components analysis with edge detection algorithms.. Dealing with Eigen faces, Eigen values and recognition rates, we plotted the corresponding graphs that are obtained using MATLAB 7p0. By reducing the dimension of the covariance matrix, we could achieve faster implementation of the algorithm, which is brought down to a few seconds over a database of 200 images. By using the canny edge detection algorithm it is found that recognition rate is improved with less matching time. It is also possible in enhancing the reduced dimensional images without filtering artifacts. This can be accomplished by wavelet principal components analysis approach with Hoff field network. The scheme refers to computing principal components for a modified set of wavelet coefficients to find Eigen spectra in spectral domain and then papering the original image on to wavelet principal comments analysis. In this way the features are indirectly emphasized.

References

[1] Pantic M, Rothkrantz L, "Facial action recognition for facial expression analysis from static face images" , IEEE Transactions on Systems, Man, and Cybernetics-Part B, vol.14,no.3,pp.1449-1461, 2004.
[2] Kotsia I, Zafeiriou S, Pitas L1. "Texture and shape information fusion for facial exp ression and facial action unit recognition", Pattern Recognition, vol.41,no.3,pp.833-851, 2008.
[3] Jin Hui, GaoWen. " The human facial combined expression recognition system", Chinese Journal of Computers, vol.23,no.6,pp. 602-608, 2000.