

## Face Recognition using Dynamic Inputs: A Review

Ms. C.B. Tatepamulwar<sup>1</sup> and Dr. V. P. Pawar<sup>2</sup>

School of Computational Sciences,

Swami Ramanand Teerth Marathwada University, Nanded (M.S.) India

<sup>1</sup>fadewarch@gmail.com <sup>2</sup>vrushvijay@yahoo.co.in

### Abstract:

Ideally a face detection system should be able to take a new face and return a name identifying that person. Mathematically, what possible approach would be robust and fairly computationally economical. If we have a database of people, every face has special features that define that person. One person may have a wider forehead, while another person has a scar on his right eyebrow from a rugby match as a young tuck. One technique may be to go through every person in the database and characterize it by these small features. Possible approach would be to take the face image as a whole identity. The goal of this paper is to extend the face recognition system to using dynamic inputs such as those obtained from CCTV cameras. This would involve detecting and tracking faces correctly in two or more cameras, pre-processing them and computing the disparity information prior to the identification stage.



### 1. Introduction:

The human capacity to recognize particular individuals solely by observing the human face is quite remarkable. This capacity persists even through the passage of time, changes in appearance and partial occlusion. Because of this remarkable ability to generate near-perfect positive identifications, considerable attention has been paid to methods by which effective face recognition can be replicated on an electronic level. Certainly, if such a complicated process as the identification of a human individual based on a method as non-invasive as face recognition could be electronically achieved then fields such as bank and airport security could be vastly improved, identity theft could be further reduced and private sector security could be enhanced.

The eigenface technique finds a way to create ghost-like faces that represent the majority of variance in an image database. Our system takes advantage of these similarities between faces to create a fairly accurate and computationally "cheap" face recognition system[1].

Kirby and Sirovich pioneered the eigenface approach in 1988 at Brown University. Since then, many people have built and expanded on the basic ideas described in their original paper. We received the idea for our approach from a paper by Turk and Pentland based on similar research conducted at MIT.

### 2. Eigen face system:

The eigenface face recognition system can be divided into two main segments: creation of the eigenface basis and recognition, or detection, of a new face. The system follows the following general flow:

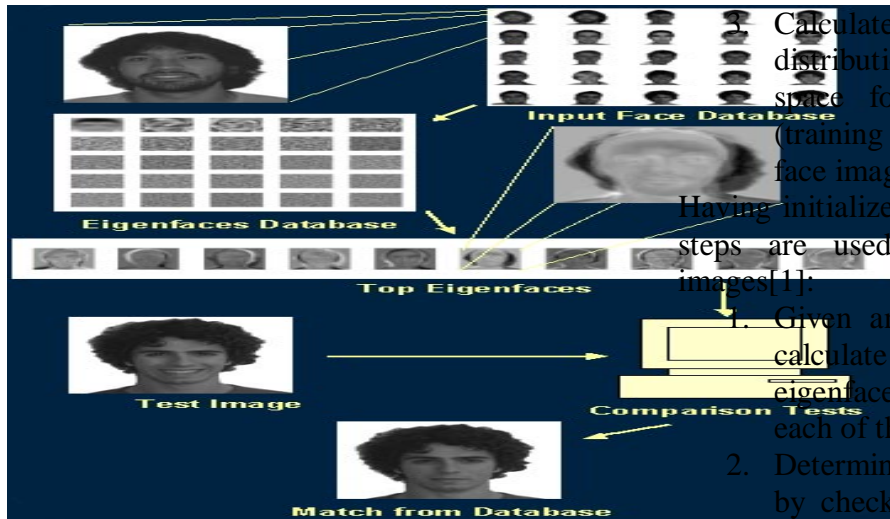


Figure 1. A robust detection system can yield correct matches when the person is feeling happy or sad(source paper: 4.)

The idea of using eigenfaces was motivated by a technique for efficiently representing pictures of faces using Principal Component analysis(PCA). It is 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 pictures. Therefore, if a multitude of face images can be reconstructed by weighted sum of a small collection of characteristic images, then an efficient way to learn and recognize faces might be to build the characteristic features from known face images and to recognize particular faces by comparing the feature weights needed to (approximately) reconstruct them with the weights associated with the known individuals.

The eigenfaces approach for face recognition involves the following initialization operations[1]:

1. Acquire a set of training images.
2. Calculate the eigenfaces from the training set, keeping only the best  $M$  images with the highest eigenvalues. These  $M$  images define the “face space”. As new faces are experienced, the eigenfaces can be updated.

3. Calculate the corresponding distribution in  $M$ -dimensional weight space for each known individual (training image), by projecting their face images onto the face space.

Having initialized the system, the following steps are used to recognize new face images[1]:

1. Given an image to be recognized, calculate a set of weights of the  $M$  eigenfaces by projecting the it onto each of the eigenfaces.
2. Determine if the image is a face at all by checking to see if the image is sufficiently close to the face space.
3. If it is a face, classify the weight pattern as either a known person or as unknown.
4. (Optional) Update the eigenfaces and/or weight patterns.
5. (Optional) Calculate the characteristic weight pattern of the new face image, and incorporate into the known faces.

## 2.1. Calculating Eigenfaces

Let a face image  $\Gamma(x,y)$  be a two-dimensional  $N$  by  $N$  array of intensity values. An image may also be considered as a vector of dimension  $N^2$ , so that a typical image of size 256 by 256 becomes a vector of dimension 65,536, or equivalently, a point in 65,536-dimensional space. An ensemble of images, then, maps to a collection of points in this huge space[1,2,8,10].

Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principal component analysis is to find the vector that best account for the distribution of face images within the entire image space. These vectors define the subspace of face images, which we call “face space”.

Each vector is of length  $N^2$ , describes an  $N$  by  $N$  image, and is a linear combination of the original face images. Because these vectors are the eigenvectors of the covariance matrix corresponding to the original face images, and because they are face-like in appearance, they are referred to as “eigenfaces”.

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

is defined by  $\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$ . Each face

differs from the average by the vector  $\Phi_n = \Gamma_n - \Psi$ . Training set with the average face. This set of very large vectors is then subject to principal component analysis, which seeks a set of  $M$  orthonormal vectors,  $\mu_n$ , which best describes the distribution of the data. The  $k$ th vector,  $\mu_k$  is chosen such that

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

is a maximum, subject to

$$\mu_l^T \mu_k = \begin{cases} 1, & l = k \\ 0, & \text{otherwise} \end{cases} \dots\dots\dots (2)$$

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

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \dots\dots\dots (3)$$

where the matrix  $A = [\Phi_1 \Phi_2 \dots \Phi_M]$ . The matrix  $C$ , however, is  $N^2$  by  $N^2$ , and determining the  $N^2$  eigenvectors and eigenvalues is an intractable task for typical image sizes. A computationally feasible method is needed to find these eigenvectors. If the number of data points in the image space is less than the dimension of the space ( $M < N^2$ ), there will be only  $M - 1$ , rather than  $N^2$ , meaningful eigenvectors (the remaining eigenvectors will have associated eigenvalues of zero). Fortunately, we can

solve for the  $N^2$ -dimensional eigenvectors in this case by first solving for the eigenvectors of and  $M$  by  $M$  matrix—e.g., solving a  $16 \times 16$  matrix rather than a  $16,384 \times 16,384$  matrix—and then taking appropriate linear combinations of the face images  $\Phi_n$ . Consider the eigenvectors  $v_n$  of  $A^T A$  such that

$$A^T A v_n = \lambda_n v_n \dots\dots\dots (4)$$

Premultiplying both sides by  $A$ , we have

$$AA^T A v_n = \lambda_n A v_n \dots\dots\dots (5)$$

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

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

$$\mu_n = \sum_{k=1}^M v_{nk} \Phi_k = A v_n, n = 1, \dots, M \dots\dots\dots (6)$$

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

### 3. Principal component analysis:

**PCA** was invented in 1901 by Karl Pearson. PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a number of uncorrelated variables called principal components, related to the original variables by an orthogonal transformation. This transformation is defined in such a way that the first principal component has as high a

variance as possible (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to the preceding components. PCA is sensitive to the relative scaling of the original variables[1,3].

The major advantage of PCA is that the eigenface approach helps reducing the size of the database required for recognition of a test image. The trained images are not stored as raw images rather they are stored as their weights which are found out projecting each and every trained image to the set of eigenfaces obtained[5].

**3.1. Eigenface Approach:**

Mathematically, the principal components of the distribution of faces are found out using the eigenface approach. First the eigenvectors of the covariance matrix of the set of face images is found out and then they are sorted according to their corresponding eigenvalues. Then a threshold eigenvalue is taken into account and eigenvectors with eigenvalues less than that threshold values are discarded. So ultimately the eigenvectors having the most significant eigenvalues are selected. Then the set of face images are projected into the significant eigenvectors to obtain a set called eigenfaces. Every face has a contribution to the eigenfaces obtained. The best M eigenfaces from a M dimensional subspace is called “face space”. Each individual face can be represented exactly as the linear combination of “eigenfaces” or each face can also be approximated using those significant eigenfaces obtained using the most significant eigen values[6,7,8].

**3.2. Steps to Calculate Eigenfaces:**

1. Select the N images of size n X n from training database.

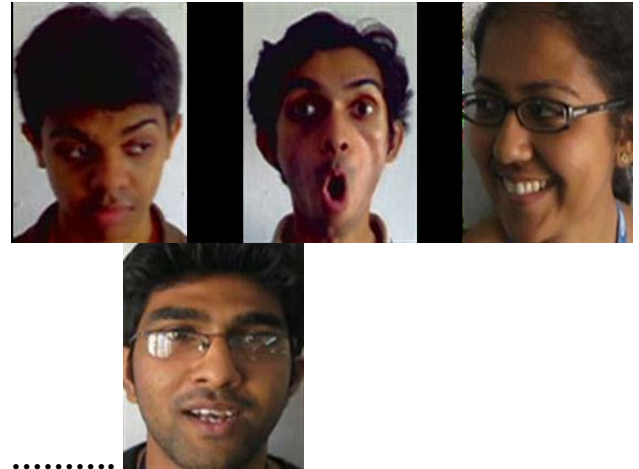


Figure. 2 Images from local Database (Training Set)[Source paper:1]

2. Calculate the mean face image for the give set.

$$I_{avg}(x,y) = \frac{1}{N} \sum_{n=1}^N I_n(x,y)$$

..... (7)



Figure. 3 Mean image of the local database (source paper: 1).

3. Calculate Zero mean image  
 $I_{zi} = I_i - I_{avg}$  where  
 $i \leq 0 < N$ .....(8)
4. Convert all the images obtained in step 3 in one dimensional vector as shown in figure below, and get a matrix  $[\emptyset]$  where  $\emptyset_i$  is one dimensional form of  $I_{zi}$

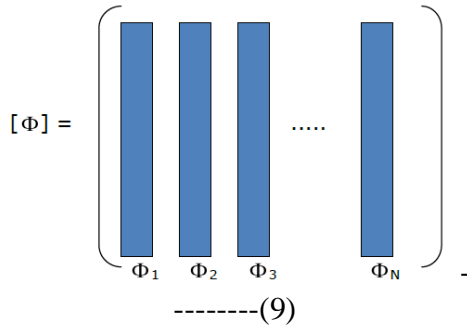


Figure. 4 Concatenated Zero mean image matrix

5. Obtain a covariance matrix, zero mean matrix A as

$$A = [\Phi^t \cdot \Phi] \dots\dots\dots(10)$$

6. Obtain Eigen value for covariance matrix [A]

$$A^x = \lambda^x \text{ and } A - \lambda_{IX} \dots\dots\dots(11)$$

Solving this we get eigen values as

$$\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \dots\dots\dots \lambda_1 \dots\dots\dots(12)$$

for each eigen value  $\lambda_i$  a associated eigen vector  $X_i$  is present. Using this  $X_i$  with matrix  $[\Phi]$  in equation 9 gives

$$\Phi \cdot X_i = F_i \dots\dots\dots(13)$$

Convert each  $F_i$  into 2 dimensional images by reserving the process of two dimension to one dimension.

7. Thus we get N eigenfaces  $F_i$  size  $n \times n$ . Following figure shows the eigenfaces for local database

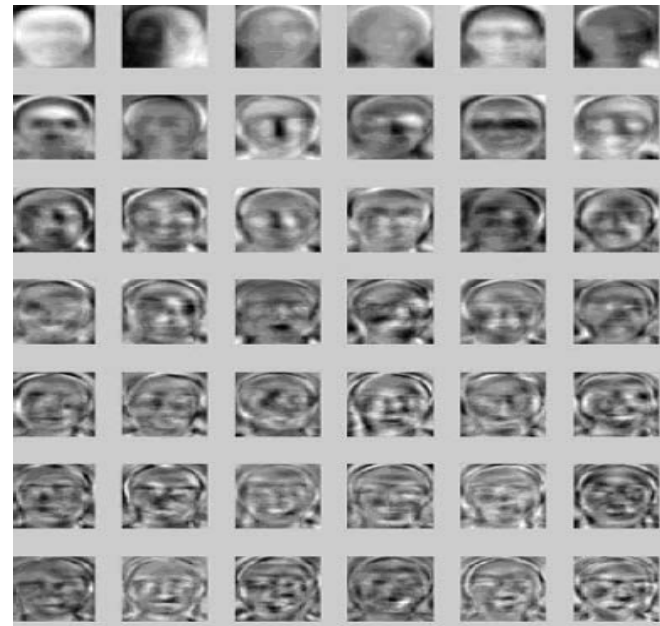


Figure. 5 Sampled Eigenfaces local databases [source 1,9]

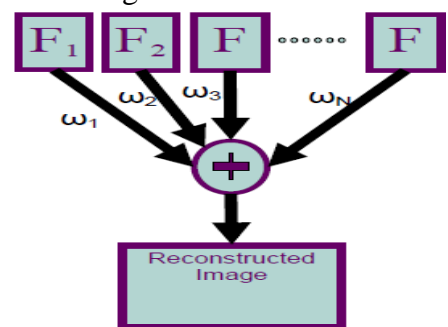
$$I_{new} = \sum \omega_i \cdot F_i + Error$$

$$where \omega_i = \frac{1}{\mu_i} \langle I_{new} | F_i \rangle$$

$$= \frac{1}{\mu_i} \sum I_{new}(x, y) F_i(x, y)$$

$$where \mu_i = \langle F_i | F_i \rangle \text{ energy in } F_i \dots\dots\dots(14)$$

Reconstructed image is obtained as shown in figure below







Original Image

Reconstructed Image

Figure. 6 Reconstruction face using eigenfaces[Source:10]

The obtained eigenfaces are orthogonal to each other.

#### 4. Video Based Face Recognition:

Recognizing face from video is probably the most difficult problem. One should be able to track the location of face, estimate the pose of face and then recognize face. In [10], their method provides the infrastructure for recognizing face from video. Face tracking and pose estimation can be done by using their method. To a face recognition system by simply adding the view based eigenspace algorithm[4,9].

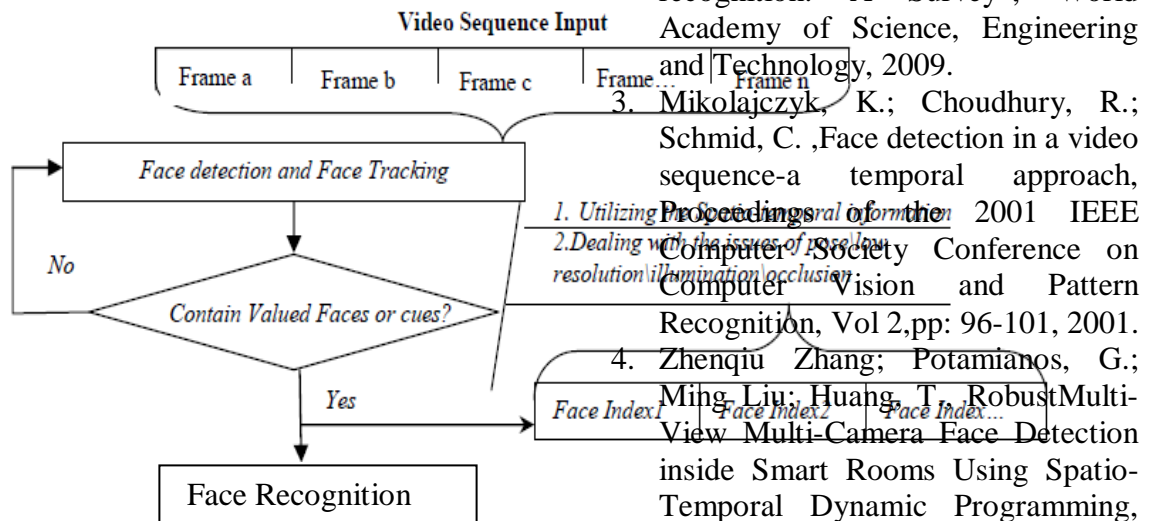


Figure 7. Scheme of Video based face recognition system(source: 9)

It is thought to have been motivated by the way humans process faces for recognition and can easily be used with dynamic inputs such as CCTV images. Such images are small and are not suitable for use with most feature based methods. Appearance based methods like Eigenfaces are appropriate for such applications [8] and its use is in line with the long term goals of this work. We are proposed Scheme of Video based face recognition system.

#### 5. Conclusion:

In this paper the face recognition based on PCA eigenfaces is studied for standard as well as locally created unconditional database. We present a comprehensive survey on video based face recognition. We have tried our best to provide researchers in the held with up-to-date information of research on video-based face recognition.

#### References:

1. HariPriya and Pinky Tejwani, "Face Recognition using Eigenfaces", ECE847 Project Report, Fall 2004, Clemson University.
2. Huafeng Wang, Young Wang and Yuah , " Video based face recognition: A Survey", World Academy of Science, Engineering and Technology, 2009.
3. Mikolajczyk, K.; Choudhury, R.; Schmid, C. ,Face detection in a video sequence-a temporal approach, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 2,pp: 96-101, 2001.
4. Zhenqiu Zhang; Potamianos, G.; Ming Liu; Huang, T. Robust Multi-View Multi-Camera Face Detection inside Smart Rooms Using Spatio-Temporal Dynamic Programming, 7th International Conference on Automatic Face and Gesture Recognition, 2-6 April 2006 Page(s):407- 412.

5. J. Yang and A. Waibel. A real-time face tracker. In Proceedings of the Third IEEE Workshop on Applications of Computer Vision, pages 142-147, Sarasota, FL, 1996.
6. G. Bradski. Computer vision face tracking for use in a perceptual user interface. Technical Report Q2, Intel Corporation, Microcomputer Research Lab, Santa Clara, CA, 1998.
7. Face Recognition using Gabor Filters, Muhammad SHARIF, Adeel KHALID, Mudassar RAZA, Sajjad MOHSIN, Department of Computer Sciences, COMSATS Institute of Information Technology, Wah Cantt-Pakistan
8. Face Detection using Neural Network & Gabor Wavelet Transform, Avinash Kaushal<sup>1</sup>, J P S Raina<sup>2</sup>, 1GCET, Greater Noida, U.P., India; 2BBSBEC, Fatehgarh Sahib, Punjab, India.
9. Heseltine, T., Pears, N., Austin, J.: Evaluation of image preprocessing techniques for eigenface based face recognition. In Proc. of the Second International Conference on Image and Graphics, SPIE vol. 4875, (2002) 677-685.
10. R. Rajlakshmi, U. Kalaimathy and M. Jayanti, "Online Voting Using Face Recognition Methods IJREAT International Journal of Research in Engineering and Advanced Technology Vol. 1 issue 5 Oct/ Nov-2013 pp.1-6