



<http://www.diva-portal.org>

This is the published version of a paper published in *International Journal of Advanced Research In Computer Science and Software Engineering*.

Citation for the original published paper (version of record):

Bhattacharyya, S., Chakraborty, S. (2014)

Reconstruction of Human Faces from Its Eigenfaces

International Journal of Advanced Research In Computer Science and Software Engineering, 4(1): 209-215

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:oru:diva-80539>



Reconstruction of Human Faces from Its Eigenfaces

Subhajit Bhattacharyya

ECE Department

Mallabhum Institute of Technology- WB, India

Subham Chakraborty

CSE Department

Mallabhum Institute of Technology- WB, India

Abstract: *Eigenface or Principal Components Analysis (PCA) methods have demonstrated their success in face recognition, detection and tracking. In this paper we have used this concept to reconstruct or represent a face as a linear combination of a set of basis images. The basis images are nothing but the eigenfaces. The idea is similar to represent a signal in the form of a linear combination of complex sinusoids called the Fourier Series. The main advantage is that the number of eigenfaces required is less than the number of face images in the database. Selection of number of eigenfaces is important here. Here we investigate what is the number of minimum eigenface that is required for faithful production of a face image.*

Keywords : *Face reconstruction, Eigen faces, Eigen vectors, Principal component analysis (PCA), Fourier Series.*

I. INTRODUCTION

This paper is a step toward developing a face reconstruction system which can reconstruct any face of a face database faithfully. This system can be useful in any visual information seeking purpose. Suppose we have a face database of 60 students of a class. These face images have high dimensionality. Instead of storing all these 60 high dimensional faces in the computer it's better to consider only a subspace with lower dimensionality to represent this face space, from which we can reconstruct any face in the database faithfully. This lower dimensional face space is called eigenfaces. In addition to this dimensionality reduction advantage we need to store selective number of eigenfaces which is obviously less than the original number of faces in the database. The scheme is based on an information theory approach that decomposes face images into a small set of characteristic feature images called 'Eigenfaces' [1], which are actually the principal components of the initial training set of face images. Reconstruction is performed by multiplying eigenfaces with weight vector. This weight vector differs for different face reconstruction. Each weight vector is nothing but a single column matrix and number of elements within this single column matrix is equal to the number of eigenfaces we select. The Eigenface approach gives us efficient way to find this lower dimensional space. Eigenfaces [2] are the Eigenvectors which are representative of each of the dimensions of this face space and they can be considered as various face features. Any face can be expressed as linear combinations of the singular vectors of the set of faces, and these singular vectors are eigenvectors of the covariance matrices. So this problem of face reconstruction is solved here by appearance based subspace analysis and it is the oldest method which gives promising results. The most challenging part of such a system is finding an adequate subspace.

When using appearance-based methods, we usually represent an image of size $n \times m$ pixels by a vector in an $n \times m$ dimensional space. These $(n \times m)$ dimensional spaces are too large to allow robust and fast object recognition. A common way to attempt to resolve this problem is to use dimensionality reduction techniques. In this paper we make use of PCA dimensionality reduction method. The organization of the paper can be summarized as: The motivation behind this paper is explained in Section-II. The Eigenface approach is briefly explained in Section-III. Face reconstruction system discussed in Section-IV. Experimental results are shown in Section-V. The Significance of Eigen face approach is explained in Section-VI. Conclusions & Future work explained in Section-VII.

II. MOTIVATION AND APPROACH

Eigenfaces has a parallel to one of the most fundamental ideas in mathematics and signal processing – The Fourier Series [8]. This parallel is also very helpful to build an intuition to what Eigenfaces (or PCA) sort of does and hence must be exploited. Hence we review the Fourier Series in a few sentences. Fourier Series are named so in the honor of Jean Baptiste Joseph Fourier who made important contributions to their development. Representation of a signal in the form of a linear combination of complex sinusoids is called the Fourier Series. What this means is that you can't just split a periodic signal into simple sines and cosines, but you can also approximately reconstruct that signal given you have information how the sines and cosines that make it up are stacked. More Formally - Put in more formal terms, suppose $f(x)$ is a periodic function with period 2π defined in the interval $c \leq x \leq c+2\pi$ and satisfies a set of conditions called the Dirichlet's conditions then $f(x)$ can be represented by the trigonometric series

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \dots\dots\dots (1)$$

The above representation of $f(x)$ is called the Fourier series and the coefficients a_0 , a_n and b_n are called the Fourier coefficients and are determined from $f(x)$ by Euler's formulae. An example that illustrates (1) or the Fourier series is:

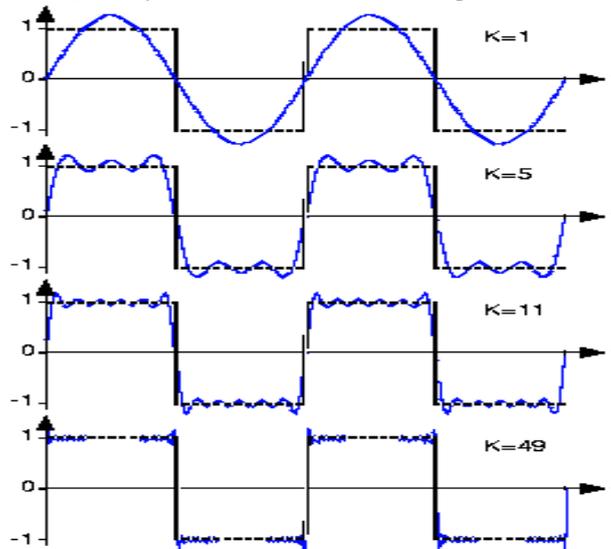


Fig. 1 A square wave (given in black) can be approximated to by using a series of sines and cosines (result of this summation shown in blue). Clearly in the limiting case, we could reconstruct the square wave exactly with simply sines and cosines.

Though not exactly the same, the idea behind Eigenfaces is similar. The aim is to represent a face as a linear combination of a set of basis images (in the Fourier Series the bases were simply sines and cosines). That is :

$$\Phi_i = \sum_{j=1}^K (w_j u_j)$$

Where Φ_i represents the i^{th} face with the mean subtracted from it, w_j represent weights and u_j the eigenvectors. The big idea is that you want to find a set of images (called Eigenfaces, which are nothing but Eigenvectors of the training data) that if you weigh and add together should give you back a image that you are interested in (adding images together should give you back an image). The way you weight these basis images (i.e the weight vector) could be used as a sort of a code for that image-of-interest and could be used as features for recognition. This can be represented aptly in a figure as:

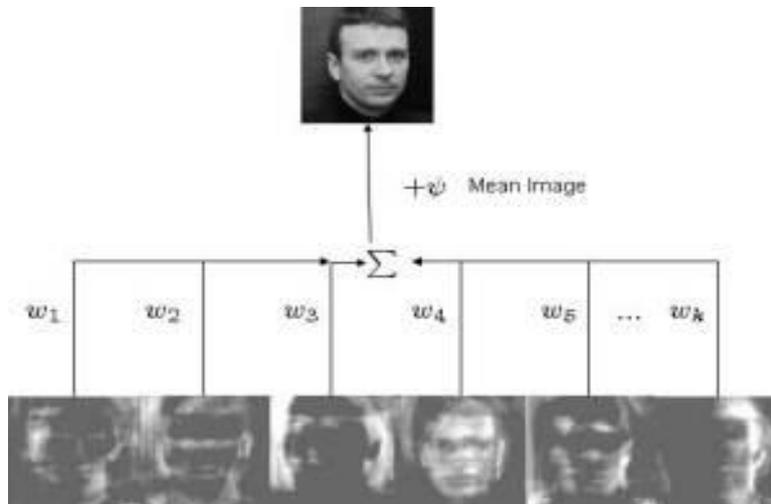


Fig. 2 Illustration of how a face image is reconstructed

In the above figure, a face that was in the training database was reconstructed by taking a weighted summation of all the basis faces and then adding to them the mean face. Please note that in the figure the ghost like basis images (also called as Eigenfaces, we'll see why they are called so) are not in order of their importance. They have just been picked randomly from a pool of 70 by me. These Eigenfaces were prepared using images from the MIT-CBCL database (also I have adjusted the brightness of the Eigenfaces to make them clearer after obtaining them, therefore the brightness of the reconstructed image looks different than those of the basis images).

III. EIGENFACE APPROACH

A. Eigen Values and Eigen Vectors

In linear algebra, the eigenvectors of a linear operator are non-zero vectors which, when operated on by the operator, result in a scalar multiple of them. The scalar is then called the eigenvalue (λ) [3] associated with the eigenvector(X). Eigen vector is a vector that is scaled by a linear transformation. It is a property of a matrix. When a matrix acts on it, only the vector magnitude is changed not the direction.

$AX=\lambda X$ (2.1) Where A is a Vector function.

The Principal Components [7] (or Eigenvectors) basically seek directions in which it is more efficient to represent the data. This is particularly useful for reducing the computational effort. To understand this, suppose we get 60 such directions, out of these about 40 might be insignificant and only 20 might represent the variation in data significantly, so for calculations it would work quite well to only use the 20 and leave out the rest. This is illustrated by this figure:

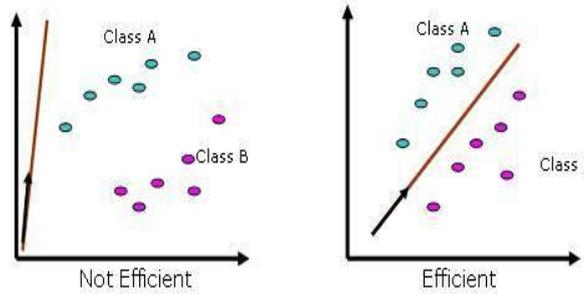


Fig. 3 Variation of data points

Such an information theory approach will encode not only the local features but also the global features. Such features may or may not be intuitively understandable. When we find the principal components or the Eigenvectors of the image set, each Eigenvector has some contribution from Each face used in the training set. So the Eigenvectors also have a face like appearance. These look ghost like and are ghost images or Eigenfaces. Every image in the training set can be represented as a weighted linear combination of these basis faces.

The number of Eigenfaces that we would obtain therefore would be equal to the number of images in the training set. Let us take this number to be M. Some of these Eigenfaces are more important in encoding the variation in face images, thus we could also approximate faces using only the K most significant Eigenfaces.

B. Calculations of Eigen Values and Eigen Vectors

By using (2.1), we have the equation,

$(A-\lambda I)X=0$ (2.2)

Where I is the n x n Identity matrix. This is a homogeneous system of equations, and from fundamental linear algebra, we know that a nontrivial solution exists if and only if $\det(A-\lambda I)=0$ (2.3)

Where $\det()$ denotes determinant. When evaluated, becomes a polynomial of degree n. This is known as the characteristic equation of A, and the corresponding polynomial is the characteristic polynomial. The characteristic polynomial is of degree n. If A is n x n, then there are n solutions or n roots of the characteristic polynomial. Thus there are n eigenvalues of A satisfying the equation,

$AX_i=\lambda X_i$ (2.4)

Where $i=1, 2, 3, \dots, n$

If the eigenvalues are all distinct, there are n associated linearly independent eigenvectors, whose directions are unique, which span an n dimensional Euclidean space.

C. Repeated Eigenvalues

In the case where there are r repeated eigenvalues, then a linearly independent set of n eigenvectors exist, provided the rank of the matrix

$(A-\lambda I)$ (2.5)

is rank n-r. Then, the directions of the r eigenvectors associated with the repeated eigenvalues are not unique.

IV. FACE RECONSTRUCTION SYSTEM

Here we have developed a system where accurate reconstruction of the face is not required what we concern here that to reconstruct a face which can be identified by human being. So,each and every details of the reconstructed face is not required. As accurate reconstruction of the face is not required, so we can now reduce the dimensionality to K instead of M($K < M$). This is done by selecting the K Eigenfaces which have the largest associated Eigenvalues. These Eigenfaces now span a K-dimensional which reduces computational time also the space.

In order to reconstruct the original image from the eigenfaces, one has to build a kind of weighted sum of all eigenfaces (Face Space). That is, the reconstructed original image is equal to a sum of all eigenfaces, with each eigenface having a certain weight. This weight specifies, to what degree the specific feature (eigenface) is present in the original image. If one uses all the eigenfaces extracted from original images, one can reconstruct the original images from the eigenfaces exactly. But one can also use only a part of the eigenfaces. Then the reconstructed image is an approximation of the original image. However, one can ensure that losses due to omitting some of the eigenfaces can be minimized. This happens by choosing only the most important features (eigenfaces).

A. Assumptions

- There are M images in the training set.
- There are K most significant Eigenfaces using which we can satisfactorily approximate a face. Needless to say $K < M$.

- All images are $N \times N$ matrices, which can be represented as $N^2 \times 1$ dimensional vectors. The same logic would apply to images that are not of equal length and breadths. To take an example: An image of size 112 x 112 can be represented as a vector of dimension 12544 or simply as a point in a 12544 dimensional space.

B. Algorithm for Finding Eigenfaces

- Obtain M training images I_1, I_2, \dots, I_M , it is very important that the images are centered.



Fig. 4 Arbitrary face database

- Represent each image I_i as a vector Γ_i as discussed above.

$$I_i = \begin{bmatrix} a_{11} & \dots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \dots & a_{NN} \end{bmatrix}_{N \times N} \xrightarrow{\text{yields}} \begin{bmatrix} a_{11} \\ \cdot \\ a_{1N} \\ \cdot \\ a_{2N} \\ \cdot \\ a_{NN} \end{bmatrix}_{N^2 \times 1} = \Gamma_i$$

- Find the average face vector Ψ .

$$\Psi = \frac{1}{M} \sum_{j=1}^M \Gamma_j$$

- Subtract the mean face from each face vector Γ_i to get a set of vectors Φ_i . The purpose of subtracting the mean image from each image vector is to be left with only the distinguishing features from each face and “removing” in a way information that is common.

$$\Phi_i = \Gamma_i - \Psi$$

- Find the Covariance matrix C

$$C = AA^T, \text{ where } A = [\Phi_1, \Phi_2, \dots, \Phi_M]$$

Note that the Covariance matrix has simply been made by putting one modified image vector obtained in one column each.

Also note that C is a $N^2 \times N^2$ matrix and A is a $N^2 \times M$ matrix.

- We now need to calculate the Eigenvectors u_i of C, However note that C is a $N^2 \times N^2$ matrix and it would return N^2 Eigenvectors each being N^2 dimensional. For an image this number is HUGE. The computations required would easily make your system run out of memory. How do we get around this problem?

- Instead of the Matrix AA^T consider the matrix $A^T A$. Remember A is a $N^2 \times M$ matrix, thus $A^T A$ is a $M \times M$ matrix. If we find the Eigenvectors of this matrix, it would return M Eigenvectors, each of Dimension $M \times 1$, let's call these Eigenvectors v_i .

Now from some properties of matrices, it follows that: $u_i = Av_i$. We have found out v_i earlier. This implies that using v_i we can calculate the M largest Eigenvectors of AA^T . Remember that $M \ll N^2$ as M is simply the number of training images.

- Find the best M Eigenvectors of $C = AA^T$ by using the relation discussed above. That is: $u_i = Av_i$. Also keep in mind that $\|u_i\|=1$.



Fig. 5 Eigenfaces of face database in fig 3

- Select the best K Eigenvectors, the selection of these Eigenvectors is done heuristically.

C. Finding Weights

The Eigenvectors found at the end of the previous section, u_i when converted to a matrix in a process that is reverse to that in STEP 2, have a face like appearance. Since these are Eigenvectors and have a face like appearance, they are called Eigenfaces. Sometimes, they are also called as Ghost Images because of their weird appearance.

Now each face in the training set (minus the mean), ϕ_i can be represented as a linear combination of these Eigenvectors u_i :

$$\Phi_i = \sum_{j=1}^K (w_j u_j), \text{ where } u_j\text{'s are Eigenfaces.}$$

These weights can be calculated as :

$$w_j = u_j^T \Phi_i.$$

Each normalized training image is represented in this basis as a vector.

$$\Omega_i = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}$$

where $i = 1, 2, \dots, M$. This means we have to calculate such a vector corresponding to every image in the training set and store them as templates.

D. Reconstruction Task

Now we have found Ω_i where $i = 1, 2, \dots, M$ and u_j where $j = 1, 2, \dots, K (K < M)$. So by just multiplying Ω_i (where i is a number which indicates which face we want to reconstruct) and $u_j (j = 1, 2, \dots, K)$ we reconstruct the face faithfully.

V. EXPERIMENTAL RESULTS

In this thesis for implementation of techniques MATLAB 7.0.2 version is used. In that image processing toolbox is used. MATLAB® is a high-performance language for technical computing.

In our experiment we have used a face database which contains 40 number of face images. The face database contains images of males, females, with glass in eye and without glass in eye.

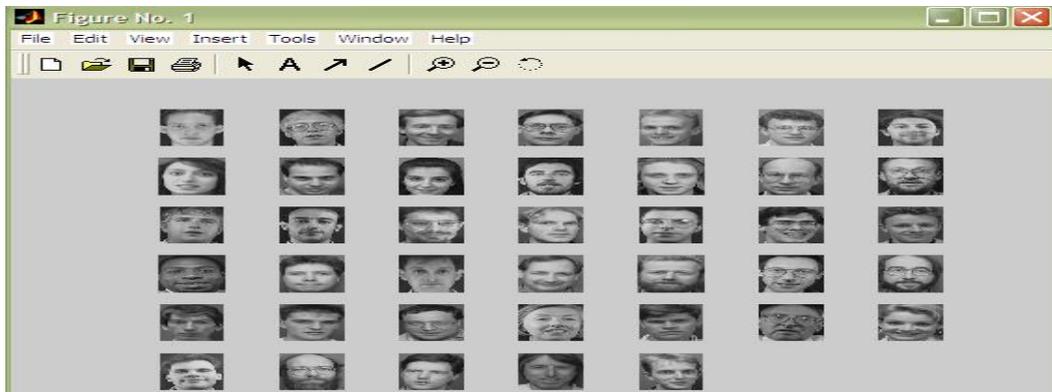


Fig. 6 Original gray scale image database

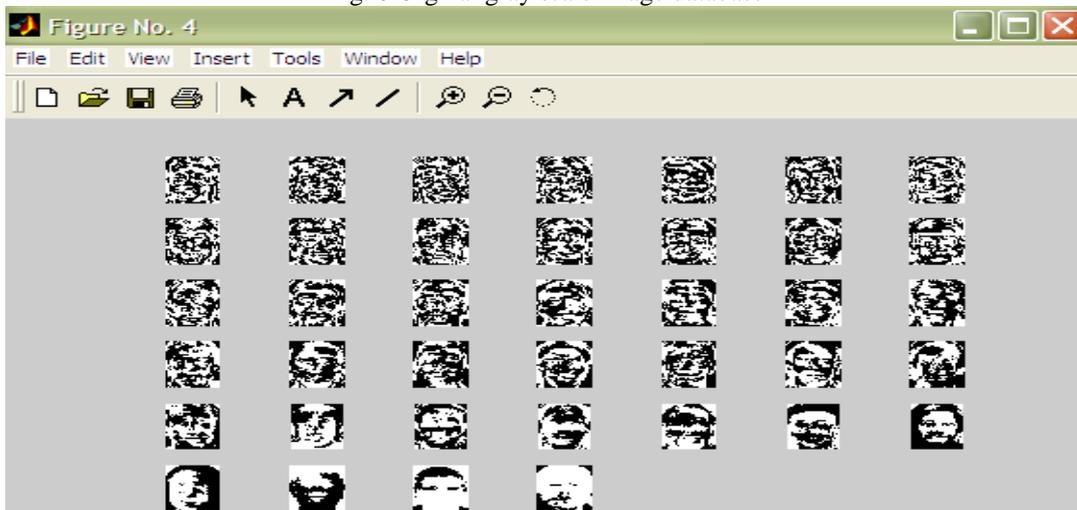
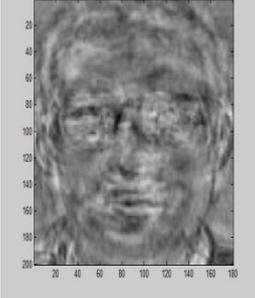
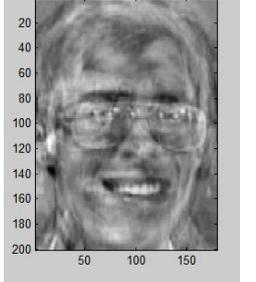
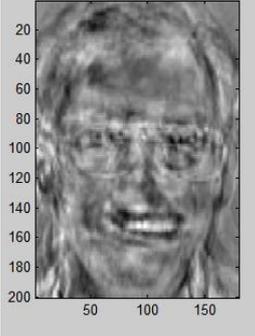
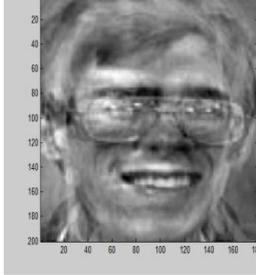
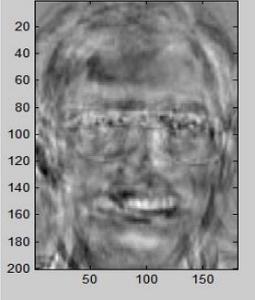
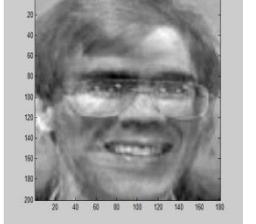
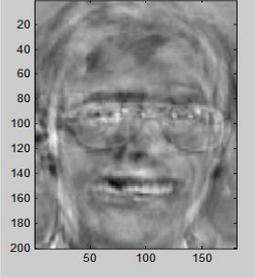
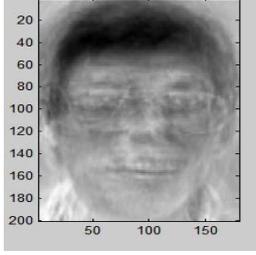


Fig. 7 Some of the eigenfaces Now we want to reconstruct the image number 20 which looks like



Fig. 8 Image number 20 in the database

TABLE- I
Number of Eigenfaces and Reconstructed image

Number of selected Eigenfaces	Reconstructed image	Number of selected Eigenfaces	Reconstructed image
5		25	
10		30	
15		35	
20		39	

So here we can easily conclude that if we choose number of eigenfaces between 30-35 we get a reconstructed image which can be easily identified by human being. Original theory states that we need to add mean face image to the summation of eigenfaes. But here we don't use the mean face addition concept but still get a reconstructed image which is easily identified with the database image.

Here we store eigenfaces and weight vectors in `eigenfaces.mat` and `weights.mat` files respectively. Before running of the main program we need to load this two files respectively. Then user have to give a number which correspond to which face is to be reconstructed.

VI. SIGNIFICANCE OF EIGENFACE APPROACH

We have seen the standard Eigenface Approach for Face Recognition. Here we find out M Eigenvectors for representing training set images. Now it is important to choose only K Eigenvectors from these M Eigenvectors, such that K is less than M , to represent face space spanned by images. This will reduce the face space dimensionality and enhance speed for face recognition. Here we are reducing the dimensionality of face images. We can choose only K Eigenvectors with highest Eigenvalues. Now as the higher Eigenvalues represent maximum face variation in the corresponding Eigenvector direction, it is important to consider this Eigenvector for face space representation. Since, the lower Eigenvalues does not provide much information about face variations in corresponding Eigenvector direction, such small Eigenvalues can be neglected to further reduce the dimension of face space. Here only K Eigenvectors with higher Eigenvalues are chosen for defining face space. The results of choosing various values for K are given in Results section. The most important point here is to choose value of K (Maximum Eigen values chosen from Co-variance Matrix) so that it does not result in high error rates for Face reconstruction process.

VII. CONCLUSIONS AND FUTURE WORKS

The Eigenface approach for Face Reconstruction process is fast and simple which works well under constrained environment. It is one of the best practical solutions for the problem of face reconstruction. Many applications which require face reconstruction do not require perfect reconstruction but a reconstructed image which can be identified by human being. So instead of searching large database of faces, it is better to keep small set of images called eigenfaces. By using Eigenface approach, from this eigenfaces any face in the database can be reconstructed faithfully. For given set of images, due to high dimensionality of images, the space spanned is very large. But in reality, all these images are closely related and actually span a lower dimensional space. By using eigenface approach, we try to reduce this dimensionality. The eigenfaces are the eigenvectors of covariance matrix representing the image space. The lower the dimensionality of this image space, the easier it would be for face reconstruction. Any image in the database can be expressed as linear combination of these eigenfaces. This makes it easier to reconstruct any image from database. We have also seen that taking eigenvectors with higher K eigenvalues instead of all M eigenvectors, does not affect performance much. So even taking lower dimensional eigenspace for the images is important here. The other important part is making this choice of K which will be crucial depending on type of application and error rate acceptable. More research needs to be done on choosing the best value of K . This value of K may vary depending on the application of Face Reconstruction. So various methods for making best choice of K needs to be studied.

REFERENCES

- [1] M.A.Turk and A.P.Pentland, "Face Recognition Using Eigenfaces", Proc. Of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, June 1991.
- [2] Matthew Turk and Alex Pentland, "Eigenfaces for Recognition", Vision and Modeling Group, The Media Laboratory, Massachusetts Institute of Technology, In J. of cognitive neuroscience, 1991, vol.3, no.1, pages 71 to 86.
- [3] Wikipedia http://en.wikipedia.org/wiki/Eigenvalue,_eigenvector_and_eigenspace
- [4] "Eigenface-based facial recognition" by Dimitri PISSARENKO
- [5] A.M. Martinez and A.C. Kak, "PCA versus LDA," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228–233, Feb. 2001
- [6] R. Gonzalez, R. Woods, S. Eddins, Digital Image Processing using MATLAB, Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2004.
- [7] 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, pp 519-524.
- [8] en.wikipedia.org/wiki/Fourier_series.