Face Detection, Recognition and Reconstruction using Eigenfaces

Sezin KAYMAK

EE574 Detection & Estimation Theory
Class Presentation
Spring 2003

Department of Electrical and Electronic Engineering
CHAPTER 1
INTRODUCTION

The face is primary focus of attention in social life. Playing a big role in conveying identity and emotion. Human ability to recognize faces is remarkable. We can recognize thousands of faces learned throughout our lifetime and identity familiar faces at a glance even after years of separation.

Computational model of face recognition are interesting because they can contribute not only the theoretical insights but also to practical applications. Computer that recognize face could be apply to a lot of wide variety of problems, including criminal identification, security systems, image and film processing, and human computer interaction. For example, to be able to model particular face distinguish it from a large number of stored models; this makes it possible to improve criminal identification.

Developing computer model of face recognition is quite difficult, because faces are complex, multidimensional visual stimuli. Face recognition is very high level task for which computational approaches are very big limited on corresponding neural activities. Previous work on face recognition tells us that there is not importance of aspect of face stimulus. Assuming predefined measurements were sufficient due to that information theory approach can help us understanding information of face images. Features may or may not relate to our intuitive notation of face features such as eyes, nose, lips, and hair etc. Our research toward developing a sort of preattentive pattern recognition capability that does not depend on having three–dimensional information or detail geometry. Our aim is to develop a computational model of face recognition that is fast, simple and accurate in limited environment such as an office or a house. Plan is based on information theory approach decomposes face images into small set of characteristics called “eigenfaces” which though of as the principle components. Approach extract relevant information in a face image, capture the variations in a collection of the face images which independent of the features encodes it as possible and compares one face with data base of models encoded similarly. By using principle component analysis to reduce the dimension of set or space so that the new basis describe the typical “models” of
the set. In our case models are a set of training faces. Components in this face space basis will be uncorrelated and maximize the variance accounted for in the original variables.

Principle component analysis aims to catch the total variation in the set of the training faces, and to explain the variation by a few variables. In fact, observation described by a few variables is easier to understand than it was defined by a huge amount of variables and when many faces have to be recognized the dimensionality reduction is important.

1.1- Background and Related Works:

There are two types of features:

1. Holistic features: Where each feature is a characteristic of whole face.
2. Partial features: Hair, nose, mouth eyes.

Holistic features which much work in computer recognition of faces has focused on detecting individual feature such as the eyes, nose mouth, and head outline, and defining face model by position, size and relation ships between these features.

Approaches are:

1. **Bledso Woodrow Wilson and Kanade’s approach**: It was the first attempt semiotamated face recognition with human computer system that classified faces on the basis of fiducal mark entered on the photographs by hand. Parameters for classification were normalized distances and ratios among points such as eye corners, mouth corners, nose tip and chin point.

2. **A.L. Yuille approach**: Improve template matching approach which measure similar features automatically, describe linear algorithm that use local template matching and global measure of fit to find and measure facial features. Strategy is based on “deformable templates” which are parameterized models of the face and its features in which the parameter values are determined by interaction with the image.
   - Disadvantages of these approaches proven difficult to extend multiple views and quite fragile.
   - Insufficient representation to account performance of human face identification but approach remains popular in computer vision.
3. **Kohenen and lahtio approaches**: Seek to capture the configurationally, or gestalt like natural of the task. Describe associative network a simple learning algorithm that can recognize (classify) Face image and recall a face image from noisy version incomplete input to the network. Use nonlinear units to train a network via back propagation to classify face images. The disadvantage of this method is there is no explicit use of configurationally properties of face and it is unclear how will scale to larger problems.

4. **T. J. Stonham’s Wisard system approach**: Is a general purpose pattern recognition device based on neural net principle applied with some success to binary face images, recognizing both identity and expression? Most of the system dealing with the input image two dimensional patterns.

5. **P. Burt. Approaches**: “Smart Sensing” within pyramidal vision machine. Approach based on multiresolution template matching use special purpose computer built to calculate multiresolution pyramid images quickly, and has been demonstrated identifying people in near real time. System work will under limited circumstances, but disadvantages, suffer difficult problems of correlation based matching, including the sensitivity to image size and noise. The face models are built by hand face images.
CHAPTER 2
EIGENFACES for RECOGNITION

Every face image can be viewed as a vector. If image width and height are $w$ and $h$ pixels respectively, the number of the components of this vector will be $w \times h$. Each pixel is coded by one vector component. The rows of the image are placed each beside one another, as shown on Figure 2.1.

Image space: This vector belong to a space, this space is the image space, the space of all images whose dimension is $w$ by $h$ pixels. The basis of the image space is composed of the following vectors. Figure 2.2

All the faces look like each other. They all have two eyes, a mouth, a nose, etc. located at the same place. Therefore, all the face vectors are located in a very narrow cluster in the image space, as shown in the Figure 2.3.
Hence, full image space is not optimal space for face description. The task presented here aims to build a face space, which better describes the faces. The basis vectors of this space are called principle components.

The dimension of the image space is $wxh$. Of course all the pixels of the face are not relevant, and each pixel depends on its neighbours. So the dimension of the face space is less than the dimension of the image space. Dimension of the face space cannot be determined but it is sure to be far less than that of the image space.

Linear algebra we want to find principle components of the distribution of the faces, or eigenvectors of the covariance matrix of the set of face images. These eigenvectors can be thought as set of features which characterize the variation between face images. Each of these images contributes more or less to eigenvector, so we can display eigenvectors as a sort of ghostly face which we call an eigenfaces; some of these faces shown in Figure 2.4.

Figure 2.3 Image and face cluster.

Figure 2.4. Visualization of eigenfaces.
Figure 2.5 shows schematically what PCA does. It takes the training faces as input and yields the eigenfaces as output. Obviously, the first step of any experiment is to compute the eigenfaces. Once this is done, the identification or categorisation process can begin.

2.1 Generation of the Eigenfaces Processes

The number of eigenfaces is equal to number of face images in the training set. However faces can also approximated using only “best” eigenfaces those have the largest eigenvalues and therefore account for the most variance within the set of face images. Reasoning for this is computational efficiency. Faces can present each face image set in terms of the eigenfaces.

Sirovich and Kirby motivated principle analysis technique, images can be approximately reconstructed by storing small collection of weights for each of face and small set of standard pictures. Images in training set can be reconstructed by weighted sums of small collection of characteristic images. So efficient way is to learn and recognize particular the faces to built characteristic features from known images and recognize particular faces by comparing feature weights needed to reconstruct them with the weights associated with known individuals.

Figure 2.6. A face developed in the face space
Figure 2.6 is showing random process that yields two dimensional result of vectors $\mathbf{x}_1, \mathbf{x}_2$. After large number of expriments of this process has been made result in Figure 2.6.

In this random process $\mathbf{x}_1$ is correlated to $\mathbf{x}_2$. It seems that some axis other than $\mathbf{x}_1$ and $\mathbf{x}_2$ more convenient to describe the process. Aim of PCA is to seek for axis that maximise the variance of data. Those that are shown Figure 2.7. They are kind of feature of the process and called feature axis.

Feature axis is orthogonal. From the graph it is obvious that the variance of the data is maximum in direction $\mathbf{p}_1$. Direction $\mathbf{p}_1$ maximises the variation of the projection of the points. The direction that yields the largest variance of the data, provided that it is orthogonal $\mathbf{p}_1$, is $\mathbf{p}_2$. Data spread is widest in direction in $\mathbf{p}_1$, next widest spread is $\mathbf{p}_2$. $\lambda_1$, the eigenvalues of the eigenvector $\mathbf{p}_1$ is 55 and $\lambda_2$, the one $\mathbf{p}_2$ is 7. %89 of
data variance explained by the first feature $p_1$ and only 11 explained by the second feature $p_2$. This means that $p_1$ capture most of the variation in original two dimensional spaces. Least important $p_2$ might be suppressed. Since this reduction of the dimension suppresses information, it should only be made if the information is not relevant for next stage of the process. Rotating the original axis by 20 degree in Figure 2.7.

Figure 2.8. Rotating the original axis by 20 degree.

In Figure 2.8 on the right, the random data in the feature space composed of the two features. On the left data are represented in a space described only by the first feature $p_1$.

Another property of the first principle component, $p_1$, is that minimizes the sum of the squared distance of the observations from their perpendicular projection onto largest principle component. Variation of the points in the direction of the second component $p_2$, is smaller than the variation of points in the direction of the largest principle component $p_1$. 
2.2 Initialization Operations in Face Recognition

The initialization operations in face recognition can be summarized in the following steps:

1. Acquire an initial set of images (the training set).

2. Calculate the eigenfaces from training set, keeping only the M images correspond to the highest eigenvalues. These eigenfaces define the face space, as new faces are experienced, the eigenfaces can be updated or calculated.

3. Calculate the corresponding distribution in M-dimensional weight space for each known individual, by projecting their face images on to the ‘face space’.

Having initialized system, the following steps are then used to recognize new face images.

1. Calculate set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the eigenfaces.

2. Determine if the image face at all (whether known or unknown) by checking to see if the image is sufficiently close to "face space".

3. If it is face, classify the weight pattern as either a known or unknown person.

4. Update the eigenface and/or weight patterns.

5. If the same unknown face is seen several times, calculate characteristic weight pattern and incorporate into known faces.

2.3 Calculating Eigenfaces:

Letting face image \( I(x,y) \) two dimensional \( N \) by \( N \) array of intensity values. Or vector of dimension \( N^2 \) if typical image of size 256x256 becomes vector of dimension 65,536 or equivalently point in 65,536 dimensional space. So image then maps to collection of points in this huge space.

Images of faces being similar in over all configurations will not randomly distributed in this huge image space and thus can be described by relatively low dimensional subspace. The main idea of principle component analysis to find eigenvectors 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 of the length of \( N^2 \). Describe \( N \times N \) image, and linear combination
of original face images. Because these vectors are eigenvectors of covariance matrix corresponding to the original face images, and because they are face-like appearance, we refer to them as “eigenfaces” some examples of eigenfaces are shown Figure 2.4.

If the training set of images be \( \Gamma_1, \Gamma_2 \ldots \Gamma_M \). The average face of the set is defined as:

\[
\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i
\]  

(2.1)

Each face differs from the average face (vector) by the vector (2.2)

\[
\Phi_i = \Gamma_i - \Psi
\]  

(2.2)

An example training set is shown in Figure 2.9(a) and average face in Figure 2.9(b). A set of large vectors is subject to principle component analysis which seek \( M \) orthonormal vector \( \mathbf{u}_k \) which best describes the distribution of the data. The \( k \)th vector \( \mathbf{u}_k \), is chosen such that:
\[ \lambda_k = \frac{1}{M} \sum_{n=1}^{M} (u_k^T \Phi_n)^2 \]  

(2.3)

The vectors \( u_k \) and \( \lambda_k \) are the eigenvectors and eigenvalues, respectively, of the covariance matrix.

\[ C = \frac{1}{M} \sum_{i=1}^{M} \Phi_i \Phi_i^T \]

\[ = AA^T \]  

(2.4)

Where the matrix \( A = [\Phi_1, \Phi_2 \ldots \Phi_M] \). The matrix \( C \), however, is \( N^2 \) by \( N^2 \) eigenvector and eigenvalues is an intractable task for typical image sizes. We need a computationally feasible method to find these eigenvectors.

If the number of data points in the image space is less than the dimension of space (\( M < N^2 \)), there will be only \( M \), rather \( N^2 \), meaningful eigenvectors. (the remaining eigenvectors will have eigenvalues of zeros.)

We can solve for \( N^2 \) dimensional eigenvectors in this case by first solving for the eigenvectors of \( M \times M \) matrix. 16x16 matrix rather than 16384x16384 matrix. Then taking appropriate linear combinations of the face images \( \Phi_i \) consider eigenvectors of \( A^T A \) such that

\[ A^T A v_i = \mu_i v_i \]  

(2.5)

Premultiplying both side by \( A \), we have

\[ AA^T A v_i = \mu_i A v_i \]  

(2.6)

From which we see that \( A v_i \) are the eigenvectors of \( (2.4) \).

Following this analysis we construct the \( M \times M \) matrix \( L = A^T A \) where \( L_{mn} = \Phi_m^T \Phi_n \), and find \( M \) eigenvectors, \( v_i \) of \( L \). These vectors determine linear combinations of the \( M \) training set face images to form the eigenfaces \( u_i \).
\[ u_l = \sum_{k=1}^{M} v_{ik} \phi_k, \quad l = 1 \ldots M \]  \hspace{1cm} (2.7)

With this analysis calculations are greatly reduced, from the order of the number of pixels in the images \( N^2 \) to order of the number of images in the training set \( M < N^2 \), and calculation become quite manageable. The associative eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizing the variation among the images.

### 2.4 Using the Eigenfaces Classify a Face Image

Eigenfaces images calculated from the eigenvectors of \( L \) span basis set with which to describe face images. (Kirby 1987) used \( M = 115 \) images and found that about 40 eigenfaces were sufficient for a very good description of the set of face images. With \( M = 40 \) eigenfaces, RMS pixel by pixel errors in representing cropped version were about \( \%2 \).

A new face image (\( \Gamma \)) is transformed into its eigenfaces components (projected into “face space”) by a simple operation.

\[ w_k = u_k^T (\Gamma - \Psi) \quad k = 1, 2, \ldots, M \]  \hspace{1cm} (2.8)

This describes set of point by point image multiplications and summations. Figure 2.10 shows images and their projections onto eigenface space.

Weights form a vector \( \Omega^T = (w_1, w_2, \ldots w_M) \) that describes the contribution of each eigenface in representing the input face image, treating the eigenface as a basis set for face images. The vector then may then be used in standard pattern recognition to find which of a number of predefined face classes, if any, best describes the face. The simplest method for determining which face class provides the best description of an input class \( \Omega_k \) that minimize the Euclidian distance of \( \mathcal{E}_k \)

\[ \mathcal{E}_k = \| \Omega - \Omega_k \| \]  \hspace{1cm} (2.9)

where \( \Omega_k \) is vector describing the \( k \)th face class.
A face is classified as belonging to class $k$ when the minimum $\varepsilon_k$ is below some chosen threshold $\theta_e$, that defines maximum allowable distance from face space. Otherwise the face is classified as “unknown” and optionally used to create a new class.

2.5 Using Eigenfaces to Detect Faces

By using knowledge of the face space we can recognize the presence of faces. We can detect and locate faces in single image.

Because creating vector of weights is equivalent to projecting the original the original face image onto low dimensional space, many image (most of them looking nothing like face) will project on to given pattern vector. This not problem for the system, however since the distance between the image and the face space is simply the squared distance between mean and input image $\varphi = \Gamma - \Psi$ and $\varphi_f = \sum_{i=1}^{m} w_i u_i$, its projection on to face space:

$$\varepsilon^2 = \|\varphi - \varphi_f\|^2 \tag{2.10}$$

Figure 2.10. Three images and their projections onto the face space defined by the eigenfaces of Figure 2.4.
From Figure 2.10, images of the faces not change when projected into face space, while projections of non-face images appear quite different. This basic idea is used to detect the presence of faces in scene; at every location in the image calculate the distance epsilon between the local sub-images and face space. This distance from face space is used as a measure of a “face-ness”. So resulting calculating the distance from face space at every point in the image is a “face map” $\varepsilon(x, y)$. Figure 2.11. shows image and image map. Low values (the dark area) indicate presence of a face. Minimum small dark place in face map correspond location of the face in the image. Figure 2.11.

![Figure 2.11.](image)

**Figure 2.11.** (a) Original image. (b) Face map, where low value (dark area) indicate presence of face.

### 2.6 Face Space Revisited

Faces in the training set, should lie near the face space, which describe images that are “face like”. Projection distance should be within some threshold $\theta$, i.e. $\varepsilon_i < \theta$. There are four possibilities for an input image and the pattern vector:

1. Near face space near known class.
2. Near face space but not near known class.
3. Distant from face space and near face class.
4. Distant from face space and not near a known face class. Figure 2.12. shows four options for the simple example of two eigenfaces.

In the first case an individual is recognized. In the second case, an unknown individual is present. Last two cases image is not a face image. Case three shows false positive. Due to significant distance between image and subspace of shown face. Figure 3 shows images and their projecting into face space. Figure 2.10 (a) and (b) are examples of case 1, while Figure 2.10 (c) shows case 4.
2.7. Recognition Experiment

Experiment performed here to show viability of approach. It is performed experiments with stored face images and builds a system to locate and recognize faces. Large data base of face images collected under wide variation of image conditions, then conducted several experiments to asses the performance under known variation of lightning, scale and orientation. Figure 2.9 (a) were images taken from database of over 2500 and 2.9(b) is the average of the these images. Sixteen subjects were used with all combinations of three head orientation, three head sizes or scales, and three lightening conditions.

Effects of the varying lightening, size and head orientation were investigated. Various groups of sixteen images were selected and use as the training set. In each training set there was one image of each person, all taken under same condition of lightening, image size, and head orientation? All images in database were then classified as being one of these sixteen individuals. No one was rejected as unknown.

Statistics were collected measuring accuracy between the conditions. In the case of infinite $\theta_\varepsilon$ and $\theta_\delta$.System achieved approximately %96 correct classification averaged over lightening variation, %85 correct averaged over orientation variation, and %64 correct averaged over size variation.

In the second experiment same procedure were followed, but threshold $\theta_\varepsilon$ was also varied. At low values of$\theta_\varepsilon$, only images which project very closely to the known face classes (cases 1 and 3 in figure 5) will be recognized. So that there will be few errors but many of images will be rejected as unknown. At high values of $\theta_\varepsilon$ most
images will be classified, but there will be more errors. Adjusting \(\theta_c\) to achieve \(\%100\) accurate recognition putting the unknown rates to \(\%19\) varying lightening, \(\%39\) orientation, and \(\%60\) for size. Setting unknown rate arbitrary resulted in correct recognition rates of \(\%100\), \(\%94\), \(\%74\) respectively.

Experiments show an increase performance accuracy as the threshold decreases, which mean that we can catch perfect recognition when threshold goes to zero. But most of the images rejected as unknown. Also changing lightening conditions causes errors, but performance drops with in the size change this due to under lightening changes alone the neighbourhood pixel correlation remains high, but under size changes the correlation from one image to another is quite low, so there is a need for a multiscale approach, then faces at particular size are compared with one another.

![Figure 2.13. Algorithm of Face Recognition.](image-url)
CHAPTER 3
RESULTS

The algorithm is implemented over a training set of size 18 images. Each image is in grey level, and has dimensions of 64x64. There are three subjects in the training set, one man and two women. Each subject gives 6 images, with frontal view with different gesture and face orientation. Figure 3.1. shows samples in training set. The training set is found from internet.
The eigenface algorithm firstly forms overall average image. This is the image just adding all images and dividing by number of images in training set. And the eigenvectors of covariance matrix that is formed by combining all deviations of training set’s images from average image is formed in order to apply eigenfaces algorithm. The average image of the training set is shown in Figure 3.2.
After finding overall average image, the order is to find eigenvectors of the covariance matrix. Since there are 18 images in the training set, we need to find 18 eigenvectors that are used to represent our training set. Visualization of eigenvectors is carried out simply applying a quantization that is if the found eigenvectors have components that are greater than 255 and smaller than 0 round them to 255, and 0 respectively. Result of eigenvectors or simply eigenfaces is shown in Figure 3.3 with corresponding eigenvalues.
Figure 3.3. Corresponding Eigenfaces of the training set, with the order of increasing in eigenvalues.
CHAPTER 4
CONCLUSION

System is extended to multi dimensional space; deal with a range of aspects (other than full frontal views) by defining small number of face classes for each known person corresponding to characteristics views. Reasoning about images in face space provides a means to learn and recognize new faces in ‘an unsupervised manner. When an image very close to face space but it is not classified as one of the familiar faces, it is labelled as “unknown”. Computer stores pattern vector and corresponding unknown image. If a collection of “unknown” pattern vectors cluster in the pattern space, the presence of a new but undefined face is postulated. A noisy image and partially occluded face should cause recognition performance degrade gracefully. Since the system implement auto associative memory for known faces. This evidenced by the projection of the occluded face image, Figure 3.b.

Eigenface approach base on information theory, recognition base on small set of image features that best approximate the set of known face image, not depends on intuitive notation of facial part and features. Although it is not first class, it is well fitted for face recognition, fast simple, work well in constrained environment.
REFERENCES


[5]. Matthew A. Turk and Allex P. Pentland, ”Face Recognition Using Eigenfaces”.

23
APPENDIX

MATLAB Source Code

```plaintext
% Implementation of Eigenfaces algorithm
% by Sezin KAYMAK
% open files orderly
clear all
M = zeros(64,64);
for(i = 1:18)
    f1 = fopen(num2str(i),'rb');
    fread(f1,13,'char');
    im = fread(f1,64*64,'uint8');
    for(k=1:64)
        for(m = 1:64)
            NIM(i,k,m) = im((k-1)*64+m);
        end
    end
    %M = M + NIM;
    %image(NIM);
    %colormap('gray');
    %pause
    fclose(f1);
end
% show 18 figures, each subject in a single figure
figure
hold
for( i = 1 :6)
    for(k = 1:64)
        for(l =1:64)
            x(k,l)=NIM(i,k,l);
        end
    end
    subplot(3,2,i),image(x);
end
colormap('gray');
figure
hold
for( i = 7 :12)
    for(k = 1:64)
        for(l =1:64)
            x(k,l)=NIM(i,k,l);
        end
    end
    subplot(3,2,i-6),image(x);
end
colormap('gray');
figure
```

24
hold
for( i = 13 :18)
    for(k = 1:64)
        for(l=1:64)
            x(k,l)=NIM(i,k,l);
        end
    end
    subplot(3,2,i-12),image(x);
end
colormap('gray');

data = double(NIM);
%M = M/18;
%image(M);
%colormap('gray');

%average vector
G = zeros(64*64,18);
for(i = 1:18)
    for(j = 1:64)
        for(k = 1:64)
            G((j-1)*64+k,i)=data(i,j,k);
        end
    end
end
Y = zeros(64*64,1);
for(i = 1:18)
    for(j = 1:(64*64))
        Y(j,1) = Y(j,1)+G(j,i);
    end
end
Y = Y / 18;
%end of average finding
%show average image
for(k = 1:64)
    for(l = 1:64)
        x(k,l) = Y((k-1)*64+l);
    end
end
figure
image(uint8(x));
colormap('gray');
pause;

% form A
for( i = 1:18)
    A(:,i) = G(:,i)-Y;
end
%find L
L = A'*A;

%find eigenvalue and eigenvector
[V, lamda] = eig(L);

%find eigenfaces
u = zeros(64*64, 18);
for(i = 1:18)
    for(j = 1:18)
        u(:,i) = u(:,i) + V(i,j)*A(:,j);
    end
end

% check eigenfaces
for(i = 1:18)
    for(j = 1:64)
        for(k = 1:64)
            EigenFaces(j,k,i) = round(u((j-1)*64+k,i));
            if(EigenFaces(j,k,i) < 0)
                EigenFaces(j,k,i) = 0;
            end
            if(EigenFaces(j,k,i) > 255)
                EigenFaces(j,k,i) = 255;
            end
        end
    end
end

%plot eigenfaces
figure
for(i = 1:6)
    subplot(3,2,i), image(EigenFaces(:,:,i));
    xlabel(['Eigen Value = ' num2str(lamda(i,i))]);
    colormap('gray');
    pause
end

figure
for(i = 7:12)
    subplot(3,2,i-6), image(EigenFaces(:,:,i));
    xlabel(['Eigen Value = ' num2str(lamda(i,i))]);
    colormap('gray');
    pause
end

figure
for(i = 13:18)
    subplot(3,2,i-12), image(EigenFaces(:,:,i));

xlabel(['Eigen Value = ' num2str(lamda(i,i))]);
colormap('gray');
pause
end

% calculat weight
w=zeros (18,18);
for (i=1:18)
    for(j=1:18)
        w(i,j)=u(:,j)'*(G(:,i)-Y);
    end
end