A Least-Squares Eigeneyes Decomposition Algorithm for Face Recognition

Mudassir Masood [1] , Mohamed Deriche [1]

[1]Department of Electrical Engineering, King Fahd University of Petroleum & Minerals.
Email: {mudassir, mderiche}@kfupm.edu.sa

Abstract

In this paper, a pose invariant face recognition system using PCA with eyes images is proposed. This system automatically detects the face from the input image. The eyes region is then extracted from the detected face image, followed by an eigen-decomposition stage. We then use such decomposition to identify test images. A new classification based on the least squares estimate to improve recognition accuracy is also proposed. Good improvement in classification is achieved outperforming the traditional PCA approach.

Keywords: PCA, ACM, wavelet, least squares.

1. Introduction

Face recognition is widely used in biometrics systems. Although some reliable biometric identification systems are present, most exhibit the drawback of subject involvement. Face recognition, on the other hand, is effective without subject involvement. To develop robust face recognition systems, a good amount of research effort has been dedicated to the problem [1]. Most of the existing systems, however, require long training and are computationally expensive. Additionally, most of the developed techniques use the complete face image [2] while few perform operations on local features like eyes, nose, mouth etc [6], [7]. It has been observed experimentally that good recognition rate can also be achieved by using only local facial features. Unlike other facial features that change with time, eyes do not change significantly. So we propose here to use the eyes for recognition. This also helps in reducing the overall computational complexity.

In this paper we propose a novel face recognition system. This system first detects the face from the input image. The system then detects the eyes. These eyes images are then separated from the face image to perform eigenfeature extraction. Such features are then used in classification. A best fit of the projection of an unknown image on eigenspace is calculated using least squares. To test the proposed system, we carried our experiments using the AR and the AT&T faces databases.

In the next section, the approach for face detection is explained followed by the extracting of the eyes images in section 3. In section 4 and 5, we describe the proposed algorithm with a discussion of the proposed least squares approximation. Experimental results are provided in section 6 followed by section 7, the conclusion.

2. Detection of the Face Region

A robust face recognition system must be able to detect faces accurately from input images. Such system must also be able to handle the different variations in the input face images. The input images can vary in size, orientation, etc. This brings in the importance of a system which can detect face regions of different sizes and orientations. Several methods have been proposed in the literature for this purpose. Here, we propose to use the Active Contour Models (ACM), also known as Snakes [12,13]. Using Snakes we can detect face regions with different orientations, sizes, expressions and occlusions.

2.1. Active Contour Models

The ACMs belong to the general category of deformable models that use energy minimization. From any starting point, the given "snake" deforms into alignment with the nearest salient features of the image. These features correspond to the local minima in the energy generated by processing the image.

More precisely we can explain the snake as a parametric contour x that deforms over a series of iterations. The contour depends on two parameters:

$$x(s,t) = \begin{cases} s = \text{space parameter} \\ t = \text{time parameter} \end{cases}$$

The shape of the contour is controlled by internal contour forces, external constraints, and image potential energy. The energy of the contour model $E(x(s))$ is defined mathematically as:

$$E(x(s)) = \int_0^1 P(x(s))ds + \int_0^1 \alpha(s) \mid x_s(s) \mid^2 \; ds + \int_0^1 \beta(s) \mid x_{ss}(s) \mid^2 \; ds \qquad (1)$$

Where P is the potential energy, $\alpha(s)$ and $\beta(s)$ are the normalization factors controlling the first and second-order derivatives respectively. The minimization of this energy can be performed using the gradient-descent algorithm. More details about Snakes can be found in [9].

2.2. Face Region Detection

The ACM is used to detect the face region as outlined in Alg. 1. The initial contour is defined as a circle around the face.

Algorithm 1 Face Detection

```
ReadImage I
m=rows(I)
n=columns(I)
Center (cm, cn)=(m/2,n/2)
radius=min(m,n)
InitializeContour (x,y)=Circle(radius,cm,cn)
for i=1 to k do
    (x,y)=DeformContour(x,y)
end for
(x,y)=ConvexHull(x,y)
ExtractImageArea I(x,y)
```

An example of an input image and the detected face boundary is shown in Fig. 1a and 1b.

(a) (b) (c)

Fig. 1: Face boundary detection.

Note, however, that there are some sharp edges and concavities in the detected face boundary. This problem can be solved by calculating the convex hull of the boundary as described in [8]. The resulting face region is shown in Fig. 1c.

2.3. Pose Estimation

In order to detect the eyes from the face image, we first need to "straighten" the face. This requires the detection of the pose. In this work we are only concerned with on-the-plane rotations.

Since faces are rarely upside down, we put a restriction on our algorithm to work only with images that are rotated within a range of 180^o i.e. from 90^o clockwise to 90^o counterclockwise. For every face image, we generate a binary mask (using the detected face) (see Fig. 2).

(a) (b)

Fig. 2: Pose estimation of faces.

This mask will obviously be of elliptical shape. We use principal component analysis (PCA) to find the major directions in the image pixels. The principal components for a typical face mask are shown in Fig. 2a. Because of the natural elliptical shape of the face, the minor axis is approximately in the direction of eyes. Such information is used to rotate the original image to the upright position. Once the image is straightened, it is easy to detect the eyes regions.

3. Extraction of Eyes Images

The next step towards face recognition is to extract the eyes images from the detected face images. We applied this technique to all 400 images to create a local eyes database. This process is described below: (see also Fig. 3)

A. The Wavelet Decomposition

The wavelet decomposition has been widely used for facial feature extraction. The main rationale being the power of wavelets in providing a good representation of edge information. The wavelet decomposition is usually performed by applying a set of filters.

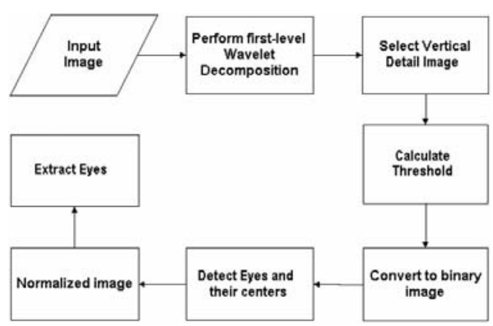

Fig. 3: Flowchart for extracting the eyes images.

By applying a low-pass (H) and a high-pass (G) filter, one gets an approximation and the details of the given image. The selection of these filters is an important issue. Here, we have used the following filters (shown to be efficient for facial feature extraction [5]):

$$H(z) = 0.853 + 0.377(z + z^{-1}) - 0.111(z^2 + z^{-2})$$
$$-0.024(z^3 + z^{-3}) + 0.038(z^4 + z^{-4})$$
$$G(z) = -z^{-1}H(z^{-1})$$

In our experiments, we have used only one-level decomposition to get an approximation and three details. Since all facial features appear in the horizontal direction, we selected the image obtained by applying high-pass filtering in the vertical direction and low-pass in the horizontal for further processing.

B. Thresholding the Edge Image

The detail image from the previous stage is then converted to a binary image by thresholding. The threshold used is the average of the highest and the lowest gray levels of the detail image. The resulting image only highlights the boundaries of the eyes, mouth, nose etc. Some other edges will also be visible depending upon the quality of the image. A morphological opening operation is performed to remove the isolated pixels (see Fig. 4).

Fig. 4: Binary image after thresholding.

C. Detecting the Eyes and their Centers

It is obvious that the position of these eyes can easily be determined from the resulting binary image. We can also separate both eyes and estimate the centers by calculating the centroids of these eyes regions.

D. Normalizing the Image

Once we have the centers of the eyes, we normalize each image such that the line between the two centers is horizontal (through rotation). Let the distance between the centers be d. After rotation, we use this distance to extract a useful area. Upon numerous observations, we formulated this area to be $1.8d$ wide and $0.65d$ high. The details are shown in Fig. 5. This is done with all images. These images are then resized to standard 30×75 images to create the database.

Fig. 5: Extraction of eyes area.

4. Recognition using Eigenfeatures

This process involves the use of principal component analysis (PCA). From the 400 images we selected 160 (4 per person) for training purposes. The remaining images were used for testing. Before training, we compute the mean image and subtract it from all images. This simplifies the implementation of PCA. In PCA, we first convert the two dimensional images into vectors by concatenating the rows. Let x_i denote the i th image in vector form of dimension $N \times 1$, then the estimated mean image (for a database of M images), is calculated as:

$$m_x = \frac{1}{M}\sum_{i=1}^{M} x_i \qquad (4)$$

Let e_i and λ_i be respectively the eigenvectors and corresponding eigenvalues of C_x (the estimated covariance matrix) given by:

$$C_x = \frac{1}{M}\left[\sum_{i=1}^{M} x_i x_i^{'}\right] \qquad (5)$$

A transformation matrix, A, whose rows are the eigenvectors of C_x can now be used to project the images into the eigenspace formed by the eigenvectors e_i s. This transformation is: $y = A(x - m_x)$. We can select a reduced number of significant eigenvectors for projecting an image into the eigenspace. As the number of eigenvectors is increased a better representation of the image is achieved. The overall recognition rate for the PCA decomposition is usually displayed as a function of the number of eigenvectors. We call the eigenvectors of faces the eigenfaces, and those of the eyes the eigeneyes.

Once we trained the system, we tested it on a number of images. Obviously, these test images were not used in the training process. These test images (after removing the mean) were projected onto the eigenspace. Similarly all training images in the database are projected onto the eigenspace. We then computed the distance between the projection of the unknown (test) image and all the projections of the training images from database. The class giving the minimum distance was declared as the class of the unknown test image. The Euclidean distance metric is used in this work. The first four eigeneyes can be seen in Fig. 6. In the next section, we discuss the method of least squares which we use to improve the recognition accuracy.

5. Improving PCA using Least-Squares (LS)

A new method based on the least squares estimate of the unknown image projection is used to improve classification. In this method, the process requires a model that relates the unknown image to the images in each class. This can be done using linear prediction. The image obtained using these coefficients is called the fitted image. To avoid the complexity that occurs due to the large size of image we use the projections of the images onto the eigenspace. The difference between the projection of the unknown image (x) and the projection of the fitted image ($\hat{x}$) is defined as the error (ε), also called residual.

$$\varepsilon = x - \hat{x} \tag{6}$$

Note that the images x and $\hat{x}$ are in vector form. The LS method minimizes the summed square of residuals to obtain the coefficient estimates. Lets say we decide to use only three training images per class, then we need three coefficients ($\alpha_1, \alpha_2, \alpha_3$). These can be found using the following equation:

$$x = [\, x_1 \, x_2 \, x_3 \,] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \varepsilon$$

$$= X_c \alpha + \varepsilon$$

Here X_C is a matrix unique for each class, formed by the projections of three images for that class and ε is the error vector. The LS estimate of the coefficient vector, $\hat{\alpha}$ can be found using:

$$X_C^T X_C \hat{\alpha} = X_C^T x$$

$$\hat{\alpha} = (X_C^T X_C)^{-1} X_C^T x$$

This $\hat{\alpha}$ is used to find $\hat{x}$: $\hat{x} = X_C \hat{\alpha}$

The estimated images ($\hat{x}$) are calculated for all classes for the given test image. After that, we evaluate the Euclidean distance between the test image and estimated images. The class of the estimated image that gives minimum distance is said to correspond to the class of the unknown test image.

Fig. 6: The first four eigeneyes.

6. Experimental Results

6.1. Face Detection

In our experiments, we use the AR face database for face detection as it contain the face and shoulder images. We generated four new databases by modifying the original database through rotation, translation and scaling. We selected a total of 600 images of 60 subjects (31 men, 29 women) from the AR database for experiments. Ten images per subject with different occlusions and illumination conditions were selected. A face was assumed to be detected accurately only if it contains the main facial features like eyes, mouth and nose. We achieved a face detection accuracy of **98.33%**. In a limited number of cases, the boundary of the faces was not well-detected. In some regions, the face boundary disappeared because of the similarity between the background shade and the skin color caused by illumination.

6.2. Eyes Detection

The results for eyes detection are shown in Table 1. The technique was applied for the AR and the AT&T databases. The proposed technique is compared with the template matching technique from the literature. We found that the performance of our method is superior to that of template matching. We considered a threshold of 15 pixels for correct detection of the eyes. We declare a point to be an eye center if the distance between the detected eye center and the original eye center is less than this threshold

	Template matching	Wavelet-based
AR (1200 eyes)	87.33%	92.33%
AT&T (800 eyes)	82%	85.75%

Table 1: Eyes detection results.

6.3. Face Recognition

In the AT&T face database, there are ten different images for each of the 40 distinct subjects. The eyes database was created using this face database. The experiments for face recognition were performed for two different numbers of training samples.

The results show that as we increase the number of eigeneyes used for classification the recognition rate improves. The recognition rate can also be increased if we increase the number of training samples. Fig. 7 compares the results when the system is trained with four and six eyes per person. The results show that the improvement in recognition rate is very good. Using the least squares method to classify an unknown (test) image, we can further improve the recognition rate for the eyes database. This is shown in Fig. 8, where we have used the least squares estimate of the unknown image.

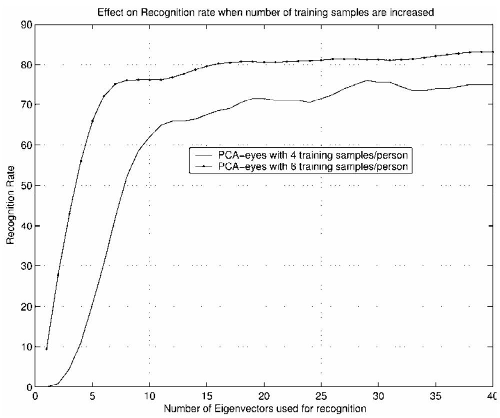

Fig. 7: Comparison of recognition rate (different number of training samples).

The work in [4] discussed the face recognition problem using only eyes. The authors used the same AT&T face database for this purpose. Both eyes of a face are extracted separately using manual location of the eyes centers. Using the minimum distance classifier of city block distance, they were able to achieve a recognition rate of 84.4% as compared to 86% for the least-squares method proposed in this paper (without manual interference). The maximum recognition rate achieved for this database was 96% reported by [3]. This was achieved, however, using the whole face images (computationally more expensive).

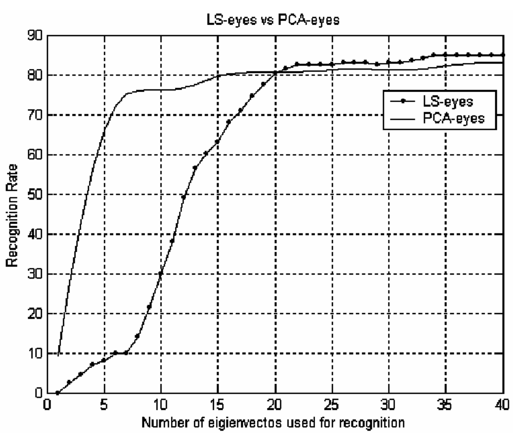

Fig. 8: Comparison of the LS-PCA and traditional PCA

7. Conclusions

For a database with large variations, we were able to get good results using only the eyes images. The recognition rate can be increased significantly if we use more samples for training. The least squares method proved to be very robust. The proposed technique is rotation, scale, and translation invariant and has the huge advantage of low computational complexity given the small size images we work with.

8. Acknowledgment

The authors would like to thank the support provided by KFUPM to carry this research.

9. References

[1] W. Zhao, R. Chellapa, P.J. Phillips and A. Rosenfeld, Face Recognition: A Literature Survey, ACM Computing Surveys, Vol. 35, No. 4, 4 December 2003, pp. 399-458.

[2] Turk, M.A.; Pentland, A.P., Face recognition using eigenfaces, Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on, 3-6 Jun 1991. Pages:586-591.

[3] S.M. Lucas, Face Recognition with the continuous N-tuple classifier, The British Machine Vision Conference, 1997.

[4] E. Hjelmas and J. Wroldsen, *Recognizing Faces from the Eyes Only*, Proceedings of the 11th Scandinavian Conference on Image Analysis, 1999.

[5] C. Garcia and G. Tziritas, *Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis*, IEEE Transaction on Multimedia, vol 1, No. 3, Pages: 264-277, March 1999.

[6] T. Kanade, *Computer Recognition of human faces*, Birkhauser, Basel, Switzerland, and Stuttgart, Germany.

[7] M. D.Kelly, *Visual identification of people by computer*, Tech rep AI-130, Stanford AI project, Stanford, CA.

[8] Barber, C. B., D.P. Dobkin, and H.T. Huhdanpaa, *The Quickhull Algorithm for Convex Hulls*, ACM Transactions on Mathematical Software, Vol. 22, No. 4, Dec. 1996, p. 469-483.

[9] Jim Ivins, John Porrill, AIVRU, University of Sheffield, England.

[10] K.M. Lam, H. Yan, *An Analytic-to holistic approach for face recognition based on a single frontal view*, IEEE Trans. Pattern Anal. Mach. Intell. 20 (7) (1998) p. 673-686.

[11] Guo Can Feng, Pong C. Yuen, *Multi-cues eye detection on gray intensity image*, Pattern Recognition, 34 (2001), p. 1033-1046