Face Recognition Using PCA and DCT

Akrouf Samir
Sehili Med Amine
Chakhchoukh Abdesslem
Mostefai Messaoud
Laboratoire MSE
Centre Universitaire de Bordj Bou Arréridj Algeria
samir_ay11@yahoo.com

Youssef Chahir
GREYC **Laboratory** CNRS UMR 6072
Departement of Computer Science
University of Caen Lower-Normandy
France
Youssef.chahir@info.unicaen.fr

Abstract—Research in the field of face recognition knew considerable progress during these last years. Among the most evoked techniques we find those which employ the optimization of the size of the data in order to get a representation which makes it possible to carry out the recognition. For these methods, the images of faces are seen like points in a space of very great dimensions. The basic idea is to encode the initial data to pass to another space of dimensions much more reduced while preserving as much useful information.
This paper presents a hybrid method combining principal components analysis (PCA) and the discrete cosine transform (DCT).
Keywords-PCA, DCT, Face Recognition

I. Presentation of PCA **(Principal Component Analysis)**

PCA is used for identification and pattern recognition. It allows expressing the data showing the distinction between their similarities and their differences. Since the problem of pattern recognition can become increasingly difficult, in particular when the data (images) are of very great dimensions, PCA can be seen as a very powerful tool to analyze the data since it operates by reducing their dimensions in a considerable way. The other advantage of using PCA is that by reducing their dimensions, the data can be compressed without losing useful information.
PCA Algorithm was used for the first time for face recognition by Mr. Turk and A. Pentland [1] in 1991 with MIT Media Labs. It is also known under the name of eigenfaces, named according to the use of the eigenvalues and eigenvectors.
The principal idea of PCA is to extract information from the image faces, encode it as much as possible and compare it with the images treated before using the same way and stored in a data base [2]. Like any algorithm of recognition, the process of the PCA is divided into two distinct phases: a phase of training and a phase of recognition (figure 1)

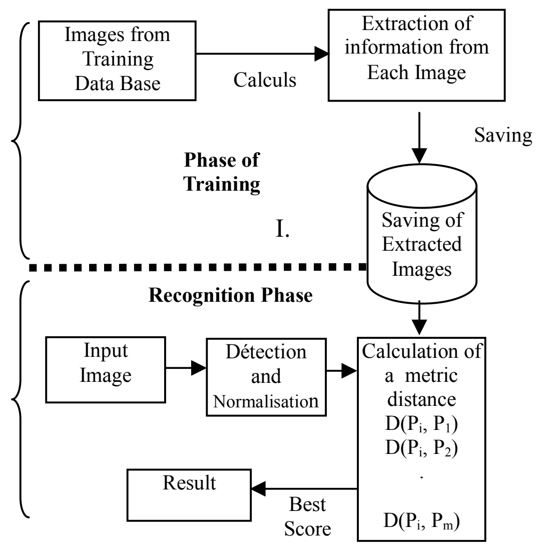

FIGURE 1: **RECOGNITION ALGORITHM STAGES.**

I.1 Training phase

In this phase, M images (for M individuals) with w width and h height will be considered as vectors of size w x h and not as matrices. The images being in gray levels, each pixel can take a value between 0 and 255. These vectors Γ_i ($1 \le I \le M$) are concatenated to make only one matrix Γ (w x h, M) of w x h lines and M columns (figure 2).

$$\Gamma = \begin{pmatrix} \Gamma_{1,1} & \Gamma_{2,1} & \cdots & \Gamma_{M,1} \\ \Gamma_{1,2} & \Gamma_{2,2} & \cdots & \Gamma_{M,2} \\ . & & . \\ & . & \\ . & & . \end{pmatrix}$$

Figure 2: Global matrix containing all the images for training

978-0-7695-3938-6/09 $25.00 © 2009 IEEE
DOI 10.1109/ICMENS.2009.48

Calculation and subtraction of the average

The average image Ψ (i.e. average vector) is calculated then withdrawn from all the images:

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \qquad (1)$$

$$\Phi_i = \Gamma_i - \Psi \ (i = 1,2,...M) \qquad (2)$$

Calculation of the covariance matrix

In this stage the covariance matrix of the data file is calculated using the following formula:

$$C = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T = AxA^T \qquad (3)$$

$$\Phi = [\Phi 1\ \Phi_2 \ldots\ \Phi_M] \qquad (4)$$

Calculation of the eigenvectors and eigenvalues

In this stage the vectors u_i and the eigenvalues associated v_I with the matrix C must be calculated. On the practical level, this calculation can sometimes be very time consuming and need a lot of memory. The matrix C is indeed a matrix of size (w x h, w x h) (resolution of the image). With a number M of images lower than the resolution (w x h), there will be only M eigenvectors which contains information (the other eigenvectors will have null eigenvalues).
To go through this problem [1] propose the following solution. For an eigenvector v_i associated to an eigenvalue λ_i we have:

$$Cv_i = \lambda_i v_i \qquad (5)$$

The matrix C has the form AA^T. Let us consider the matrix $L = A^T A$ having the eigenvectors u_i associated to eigenvalues e_i:

$$Lu_i = e_i u_i$$

$$\text{Let } A^T A u_i = e_i u_i$$

By multiplying by A the left of the two sides of the equality, we obtain:

$$AA^T A u_i = A e_i ui \qquad (6)$$

And since $C = AA^T$ we can simplify (6):

$$CAu_i = Ae_i u_i$$

$$C(Au_i) = e_i (Au_i)$$

According to the definition of the eigenvectors and eigenvalues of the matrix C we have:

$$v_i = Au_i$$

$$\lambda_i = e_i$$

The matrix L is a matrix M x M, the calculation of its eigenvectors and eigenvalues is much easier than with the matrix C. We pass indeed from a complexity about the resolution of the image to a complexity of the order of the number of images.
Then comes the stage of selection of the eigenvectors and eigenvalues where PCA selects only M' (M' < M) associated to the largest eigenvalues (those associated to the smallest eigenvalues contains only very little useful information).
Here a new vector space E_v, called face of space (facespace), is generated by the M' eigenvectors selected.
The representation of the eigenvectors points out phantom images each one proposing a part of the face [2], we call them *eigenfaces* (figure 3).
In the last stage of the phase of training, the starting images are projected on the faces space; each image Γ_i is then transformed into its components eigenfaces (w_k) by a simple vectorial projection:

$$W_k = e_k^T(\Gamma_i - \Psi)\ (1 \leq K \leq M') \qquad (7)$$

The w_k are called weights and form a vector Ω^T:

$$\Omega^T = [w_1, w_2, w_3 \ldots, w_{M'}] \qquad (8)$$

Vectors Ω^T are saved in order to be used to classify a new image in the phase of recognition.

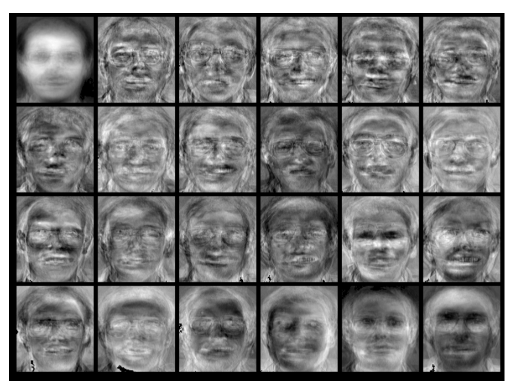

Figure 3: **Average image and the first 24 eigenfaces.**

I.2 Recognition phase

The process of assignment of a new image Γ_i to a class resulting from the training phase proceeds on two stages:
First, the image Γi is transformed into its components eigenfaces according to the formula (7).

$$\Omega^{T}_{new} = [w_1, w_2, w_3\ldots, w_{M'}]$$

Then, the class of face providing the best description of Γ_{new} is determined by calculating the minimal distance between the vector Ω^{T}_{new} and those stored in the data base. The most used metric is the Euclidean distance given by:

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (9)$$

In practice, the space of E_v faces is transformed into a space of Mahalanobis by dividing each eigenvector on the square root of the eigenvalue which is associated to it [2].
A face Γ_{new} belongs to a class K when the minimum distance between Ω_{new} and Ω_k (1<K<M) is below a certain threshold θ, otherwise the face is regarded as unknown and can be possibly used to create a new class.

I.3 Adding a new person

The addition of a new person at the base can be carried out by a simple projection of the new image on the space of E_v face. This method can be used when the data base of training is relatively large and the faces stored there are representative/.
However, it would be interesting after a long time (after several additions using the described method) to carry out the entire process of training and generate a new space of face by taking into account the new images; thus we obtain clean faces more representative of the base.

II. Presentation of DCT

DCT is a technique which owes a great part of its popularity with the field of image and video compression. It was used by Ahmed *et al.* in 1974 to transform an image signal from a space representation into a frequential representation. In 1992, the first international standard for the compression of images, JPEG, was established. It used a coder-decoder containing DCT for encoding and decoding images.
The use of DCT in face recognition field is one of the most recent methods. It uses the discrete transformation into a cosine to eliminate the redundancies in an image and extract from them the most significant elements (i.e. coefficients) in order to use them for recognition.

II.1 Training phase

An entry image is divided into blocks of size 8 X 8. Each block is transformed independently by using a two-dimensional DCT function (2d-dct):

$$F(u,v) = \frac{2}{N} c(u)c(v) \sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) \mathrm{Cos}\frac{(2x+1)u\pi}{2\pi} \mathrm{Cos}\frac{(2y+1)v\pi}{2\pi} \qquad (10)$$

Where (x, y) are the space co-ordinates of the elements of the block and (u, v) the co-ordinates of the coefficients. The term C(x) is defines by:

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}}, & for\ x = 0 \\ 1, & otherwise \end{cases}$$

We conclude that the coefficient F (0, 0) (called sometimes CD coefficient) is the most significant block because for u = 0 and v = 0 the cosine will be 1.

$$F(0,0) = \frac{1}{8}\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) \qquad (11)$$

CD Coefficient represents in fact the proportional average of the 64 elements of the block**.**
Although the total block energy of entry remains the same one in the block of coefficients, its distribution changes according to co-ordinates u and v. The transformation leads to the fact that the coefficients with the lowest frequencies (i.e. coefficients of the left higher corner) take the greatest part of energy. This property can be exploited to retain only a small number of coefficients with only few loss of useful information. It should be noted that DCT is regarded as the second best transformation after PCA in data compression.

Choice of the size of the block

The size of a block is a parameter to be chosen with precaution to obtain equilibrium between time calculation and compression rate. Blocks of significant
size make it possible to reach better data compression, whereas blocks of small size require less time computing and a higher compression rate. The choice of a size of 8 X 8 seems to offer a compromise between the two (computing time and quality of compression).

Choice of the number of coefficients

The robustness of DCT is due to the choice of a very small number of coefficients per block. The selected coefficients are those of the left higher corner of each block.

Although JPEG standard uses *a* zigzag parsing to put a block of 64 elements in a vector form (the same technique is used by [9]), this procedure is not of a great help to the algorithm of recognition because we select only one square subset of coefficients (1, 4 or 9).

Energy Histogram

The histograms are a technique usually used in numerical imagery. They were introduced by *Swain and Ballard* in 1990 via the use of a histogram of colors. A histogram consists of a number of boxes of which each one corresponds to a beach of values. A histogram of colors of an image is obtained by discretizing the colors of the image and by counting the number of occurrences of each color in the field of the values of the corresponding box.
The histograms of colors have the advantage of being invariants whatever is the rotation and the translation or the angle of sight of the image. In spite of these advantages, the histograms of colors prove sometimes that they are ineffective if the conditions of illumination change. They permit either to distinguish two images having very close color distributions with different dispersions and can suffer from the problems of calculations when the images are of very great dimensions. An energy histogram is similar to a color histogram, but instead of counting the colors of the pixels, the energy histogram calculates the occurrence of DCT coefficients in a beach of values. In comparison with the traditional histograms, the energy histograms are less resources consuming since the redundancies are eliminated by DCT calculation and dimensions are considerably reduced [9].

Training Process

We can use several images by individual to calculate an energy histogram. Since the values of pixels are between 0 and 255 and that the size of a block is 8 X 8, the minimal and maximum values which can be obtained with DCT calculation are respectively -925 and 2040.The energy histogram could then be built by withdrawing the minimum of each coefficient and aqnd then dividing by the number of boxes (bins). The following pseudo code explains the procedure:

```
static final int MIN_VALUE = -925;
int bins;/ / a number of boxes
int [ ] energyHistogram = new int [ bins ];
For each coefficient C
energyHistogram[ (C - MIN_VALUE)/bins ] ++;
```

The number of boxes is chosen experimentally and can be between 10 and 50.
To improve the recognition rate, the adjacent blocks can have a certain number of joint pixels; we then speak about a step of overlapping (*overlap*) (Figure 4). Although more expensive, this technique offers higher recognition rates.
A histogram is created for each class (i.e. individual) and stored in order to be used in the recognition phase.

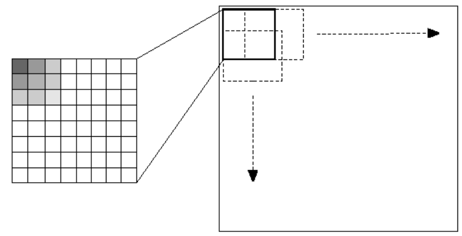

Figure 4: Only the coefficients of the left higher corner of each block are calculated.

II.2 Recognition

The energy histogram of a new entry is calculated then compared with all the histograms resulting from the phase of training by using the Euclidean distance. As the precedent algorithm, the new image is attributed to the class offering the minimal distance.

II.3 Adding a new person

The addition of a new class at the data base is extremely simplified with this algorithm. Here we just calculate the energy histogram of the new entry without taking into account the people already present in the data base (which is different in PCA).

III. Presentation of the Hybrid method

According to what precedes we note that the two methods PCA and DCT have certain mathematical similarities since that they both aim to reduce the dimensions of data. The use of a hybrid method combining these two techniques gave performances slightly higher than those obtained with only one method (experiments being made on three different image data bases). Its principle is very simple: each image is transformed into a coefficient vector (in the training and recognition phase). We first use the DCT method which produces a result used as entry for the PCA method. We use PCA with coefficients vectors instead of pixels vectors. We notice that this technique requires more time than PCA (because of the calculation of the coefficients) in particular with data bases of average or reduced size but it should be noted that it requires less memory what makes its use advantageous with bases of significant size.

IV. Experimental results

The three methods presented above were tested by using the image data bases ORL, Yale Faces and BBAFaces. The latter was created at the University Center of Bordj Bou Arreridj in 2008. It is composed by 23 people with 12 images for each one of them (for the majority of the people, the images were taken during various sessions). The images reflect various facial expressions with different intensity variations and different light sources. To facilitate the tests, the faces were selected thereafter manually in order to get images of 124 X 92 pixels, we then convert them into gray levels and store them with JPG format. Figure 5 represents a typical example of the data. It should be noted that certain categories of this data are not retained for the tests.

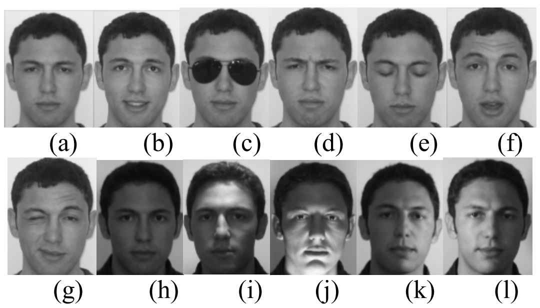

Figure (5): Exemple de la base BBAFaces. (a): normal, (b): happy, (c): glasses, (d): sad, (e): sleepy, (f): surprised, (g): wink, (h): dark, (i): top light, (j): bottom light, (k): left light, (l): right light.

Test Protocol

In order to be able to compare what was realized with other works, the three data bases mentioned earlier are used to test the performances and time execution of the methods presented. ORL is the principal data base used where the various algorithms are subjected to a bunch of tests to illustrate the parameters (number of eigenvectors, number of images used for learning, etc.) which maximizes the rate of recognition. For YaleFaces and BBAFaces, we will only quote the best performances obtained with PCA and DCT_PCA methods. It is important to note that times of training and recognition are given in milliseconds. The computer used to carry out these tests is an Intel Pentium 4 with 3 GHZ and 512 Mo of RAM.

Experimental Results

In the following we will expose the results obtained for the tests realized with YaleFaces and BBAFaces.

IV.1 Tests with Yale Faces

IV.1.1 Test of PCA

Test including all the images: 62%

Test by taking off images containing abrupt changes of luminosity: 77.14%

IV.1.2 Tests of the Hybrid method

Test including all the images: 68%

Test by taking off images containing abrupt changes of luminosity: 84.76%

IV.2 Tests with BBAFaces

IV.2.1 Test of PCA

Test including all images except top light, bottom light and dark: 57.06%

Test by taking off images containing abrupt changes of luminosity: 70.2%

IV.2.2 Test of the Hybrid method

Test including all images except top light, bottom light and dark: 66.30%

Test by taking off images containing abrupt changes of luminosity: 80%

4.3 Better Performances obtained (Tests with ORL)

PCA: 71.38 %

DCT: 67 %

Hybrid method: 72.77 %

Conclusion:

PCA can be regarded as a very fast algorithm with a more or less high robustness in spite of the fact that it uses one and only one image per anybody for the training phase. Its principal problem remains the variations of luminosity. The use of the space of Mahalanobis remarkably improves the rate of recognition of PCA.

The performances of DCT are almost comparable with those of PCA but with a higher computing time.

Finally we conclude that the combination of PCA with DCT offers higher rates of recognition than those obtained with only one method.

References

[1] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Science*, pages 71–86, 1991.

[2] N Morizet, Thomas Ea, Florence Rossant, Frédéric Amiel et Amara Amara, Revue des algorithmes PCA, LDA et EBGM utilisés en reconnaissance 2D du visage pour la biométrie, Tutoriel Reconnaissance d'images, MajecStic 2006 (Manifestation des Jeunes Chercheurs STIC). Institut Supérieur d'Electronique de Paris (ISEP).

[3] Lindsay, I. Smith (2002). A tutorial on principal components analysis. http://kybele.psych.cornell.edu/~edelman/Psych-465 Spring-2003/PCA-tutorial.

[4] Dimitri PISSARENKO. "*Eigenface-based facial recognition*". Dec1, 2002

[5] J. Ross Beveridge, Kai She, Bruce Draper, and Geof H. Givens. A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Hawaii, December 11–13 2001.

[6] ILKER Atalay. Face Recognition using eigenfaces. M. Sc. THESIS January, 1996. Istanbul technical university. Institute of science and technology.

[7] A. lemieux, "système d'identification de personnes par vision numérique " pour l'obtention du grade de maître des sciences (M.Sc.), Université Laval QUEBEC, décembre 2003.

[8] Fabien CARDINAUX. Face Authentication based on Local Features and Generative Models. Thèse NO 3410 présenté en 2005 à l' Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Suisse.

[9] Ronny Tjahyadi, Wanquan Liu, Svetha Venkatesh. Application of the DCT Energy Histogram for Face Recognition. 2nd International Conference on Information Technology for Application (ICITA 2004) PP 305-310

[10] Hazim Kemal Ekenel, Rainer Stiefelhagen. Local appearance based face recognition using discrete cosine transform. CVPRW 2006. on Volume , Issue , 17-22 June 2006 Page(s): 34 - 39