Eigenspace-based Face Recognition

Javier Ruiz del Solar, and Pablo Navarrete

Department of Electrical Engineering, Universidad de Chile.
Av. Tupper 2007, Santiago - CHILE
Email: {jruizd, pnavarre}@cec.uchile.cl

Abstract. Eigenspace-based face recognition is a very well known and successful face recognition paradigm. Different eigenspace-based approaches have been proposed for the recognition of faces. They differ mostly in the kind of projection method been used and in the similarity matching criterion employed. The aim of this paper is to present a survey of these different approaches. A general framework of the eigenspace-based face recognition paradigm is presented and different approaches are compared using theoretical aspects and simulations performed using a face database.

1 Introduction

Among the most successful approaches used in face recognition we can mention *eigenspace-based* methods, which are mostly derived from the *Eigenface*-algorithm [5]. These methods project the input faces onto a dimensional reduced space where the recognition is carried out, performing a holistic analysis of the faces. Different eigenspace-based approaches have been proposed. They differ mostly in the kind of projection/decomposition method been used, and in the similarity matching criterion employed. The aim of this paper is to present a survey of these different approaches. Three different projection methods (Principal Component Analysis, Fisher Linear Discriminant and Evolutionary Pursuit) and five different similarity matching criteria (Euclidean-, Cosine- and Mahalanobis-distance, Self-Organizing Map, and Fuzzy Feature Contrast) are here considered.

The article is structured as follows. A general framework of the eigenspace-based face recognition paradigm is presented and different eigenspace-approaches are described in section 2. In section 3 these different approaches are compared using the Yale University - Face Image Database. Finally, in section 4 some conclusions of this work are given.

2 Eigenspace-based Approaches

Eigenspace-based approaches approximate the vector faces (image faces) with lower dimensional feature vectors. The main supposition behind this procedure is that the face space (given by the dimension of the feature vectors) has a lower dimension than the image space (given by the number of pixels in each image),

and that the recognition of the faces can be performed in this reduced space. These approaches consider an off-line phase or training, where the face database is created and the so-called *projection matrix*, the one that achieve the dimensional reduction, is obtained from all the database face images. In the off-line phase are also obtained the so-called *mean face* and the reduced representations of each database image. These representations are the ones to be used in the recognition process.

2.1 General Approach

Figure 1 shows a block diagram of a generic eigenspace-based face recognition system. A preprocessing module performs a normalization of the input face images and then a subtraction of the *mean face* ($\bar{\mathbf{x}}$). Normalization is necessary to initialize each face vector **x** with the same energy.

After that, $\mathbf{x}-\bar{\mathbf{x}}$ is projected using the projection matrix $\mathbf{W} \in R^{N \times m}$ that depends on the eigenspace method been used (see section 2.2). This projection corresponds to a dimensional reduction of the input space, starting with vectors **x** in R^N (with N the vector image dimension), and obtaining projected vectors **q** in R^m with $m<N$ (usually $m<<N$). Depending on the eigenspace approach been used, the topology of the original face space would be preserved or not, and the reconstruction of the face vector **x** will be possible. For face recognition tasks it is not critical the reconstruction ability of the projection.

The *Similarity Matching* module compares the similarity of the reduced representation of the query face vector **q** with the reduced vectors $\mathbf{p}^k \in R^m$ that represent the faces in the database (see figure 1 again). By using a given criterion of similarity (see section 2.3), this module determines the most similar vector $\mathbf{p}^k$ in the database. The class C^k of this vector is the result of the recognition process, i.e. the identity of the face. In addition, a *Rejection System* for unknown faces is used if the similarity matching measure is not good enough. The rejection parameter of this system could be determined using the *Bayesian Optimal Criterion* proposed in [**4**].

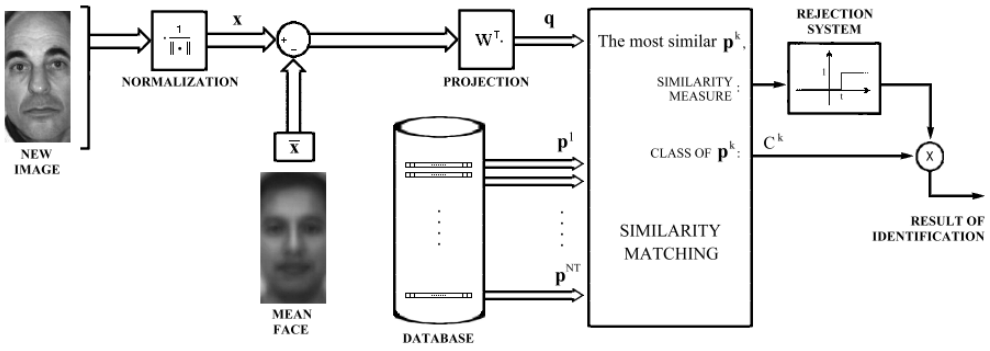

Fig. 1. Block diagram of a generic eigenspace-based face recognition system.

2.2 Projection/Decomposition Methods

2.2.1 Principal Components Analysis - PCA

PCA is a general method to identify the principal differences between signals and after that to make a dimensional reduction of them. In order to obtain the eigenfaces (face vectors in the reduced space), we first need to obtain the projection axes in which exists the largest variance of the projected face images. Then, we repeat this procedure in the orthogonal space that is still uncovered, until we realize that there is no more variance to take into account. The theoretical solution of this problem is well known and is obtained by solving the eigensystem of the correlation matrix $\mathbf{R} \in R^{N \times N}$:

$$\mathbf{R} = \mathrm{E}\left\{(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^{\mathbf{t}}\right\} \qquad (1)$$

where $\mathbf{x}$ represent the normalized image vectors, $\bar{\mathbf{x}}$ is the mean face image, and N is the original vector image dimension. The eigenvectors of this system represent the projection axes or eigenfaces, and the eigenvalues represent the projection variance of the correspondent eigenfaces. Then by sorting the eigenfaces in descendent order of eigenvalues we have the successive projection axes that solve our problem.

The main problem is that $\mathbf{R} \in R^{N \times N}$ is too big for a reasonable practical implementation. Furthermore, given NT face images as the target set, we are limited to use an estimator for the correlation matrix, by taking the averages over the target set. Let $\mathbf{X} = \left[(\mathbf{x}^1 - \bar{\mathbf{x}})(\mathbf{x}^2 - \bar{\mathbf{x}}) \cdots (\mathbf{x}^{NT} - \bar{\mathbf{x}})\right]$ be the matrix of the normalized target vectors. Then, the $\mathbf{R}$ estimator will be given by $\mathbf{R} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$. We could say that the number of eigenfaces must be less than or equal to NT, because with NT target images all the variance must be projected into the hyperplane subtended by the target images. In other words the rank of $\mathbf{R}$ is less than or equal to NT. Thereafter they could have more null or negligible eigenvalues depending on the linear dependence of the vectors in the target set. In addition, the eigensystem of $\mathbf{X}^{\mathrm{T}}\mathbf{X} \in R^{NT \times NT}$ has the same non-zero eigenvalues of $\mathbf{R}$, because its eigensystem $\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{v}^k = \lambda_k \mathbf{v}^k$ pre-multiplied by $\mathbf{X}$ gives $\mathbf{X}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{v}^k) = \lambda_k (\mathbf{X}\mathbf{v}^k)$, that represents the eigensystem of $\mathbf{R}$.

Then we can solve the reduced eigensystem of $\mathbf{X}^{\mathrm{T}}\mathbf{X} \in R^{NT \times NT}$. The correspondent eigenvalues are the same eigenvalues of the original system, the eigenfaces are represented by $\mathbf{w}^k = \mathbf{X}\mathbf{v}^k$, and to be normalized they must be divided by $\left\|\mathbf{w}^k\right\| = \sqrt{\lambda_k}$. Then, using matrix notation, the eigenface projection matrix $\mathbf{W}_{\mathrm{EF}}$ is given by $\mathbf{X}\,\mathbf{V}\,\boldsymbol{\Gamma}^{-1/2}$, where $\mathbf{V} = \left[\mathbf{v}^1 \ldots \mathbf{v}^m\right]$ and $\Gamma_{ij} = \lambda_i \, \delta_{ij}$. This procedure is shown in figure **2**.

To improve the dimensional reduction is recommendable to apply a given criterion for neglect the components with small projection variance. If we just ignore a number of components, the mean square error of the given representation is the sum of the eigenvalues not used in the representation. Therefor a good criterion would be to choose only m components, obtained by the *normalized Residual Mean Square Error* [**3**]:

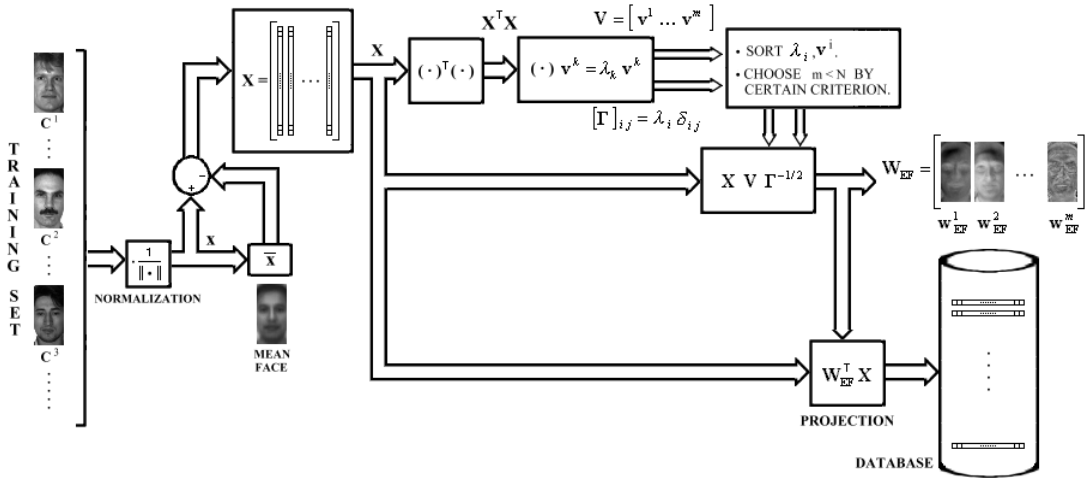

Fig. 2. PCA procedure used to obtain the projection matrix and the face projections vectors of the database. $\mathbf{W}_{EF}$ is the resulting projection matrix.

$$\mathrm{RMSE}(m) = \sum_{k=m+1}^{NT} \lambda_k \Big/ \sum_{k=1}^{NT} \lambda_k \qquad (2)$$

Considering m given by RMSE(m)<5% will be good for standard applications.

2.3 Fisher Linear Discriminant - FLD

FLD searches for the projection axes on which the face images of different classes are far from each other (similar to PCA), and at the same time where the images of a same class are close from each other. Then, in addition to the statistics used by PCA, FLD considers statistics of inner-class distributions. Where PCA seeks projection axes that are efficient for representation, FLD seeks for projection axes that are efficient for discrimination. Figure 3 shows the graphical contrast between PCA and FLD criterions in a 2D to 1D dimensional reduction example. Here two sets of normal distributed points are projected into the first PCA and FLD axes. In the PCA projection the separation between the classes is not clear because the points are mixed. In the FLD projection the separation between classes is clear enough because each class is clustered and well separated from the other class.

In order to define the FLD mathematical structure, we define the criterion function $\gamma(\mathbf{u})$ to be maximized on the successive projection axes as:

$$\gamma(\mathbf{u}) = \frac{s_b(\mathbf{u})}{s_w(\mathbf{u})} \qquad (3)$$

with $\mathbf{u}$ any projection unitary vector in the image space, and $s_b(\mathbf{u})$, the between-class scatter, $s_w(\mathbf{u})$, the within-class scatter, given by:

$$s_b(\mathbf{u}) = \sum_{i=1}^{NC} \mathrm{P}(C_i)\left\{\left(\mathbf{m}^{(i)} - \mathbf{m}\right)\mathbf{u}\right\}^2, \qquad (4)$$

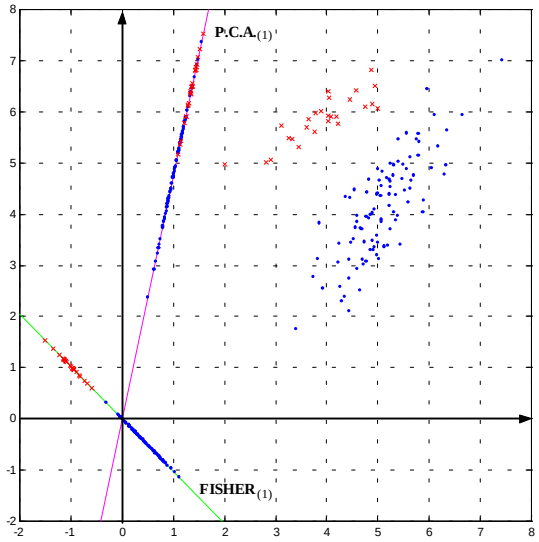

Fig. 3. PCA and FLD dimensional reduction example from 2D into 1D spaces.

$$s_w(\mathbf{u}) = \sum_{i=1}^{NC} \mathrm{P}(C_i)\mathrm{E}\left[\left\{\left(\mathbf{x}^{(i)} - \mathbf{m}^{(i)}\right)\mathbf{u}\right\}^2\right] \tag{5}$$

where *NC* is the number of classes, $\mathbf{m}$ is the global mean vector, $\mathrm{P}(C_i)$ are the probabilities associated to each class C_i, $\mathbf{m}^{(i)}$ are the average vectors of C_i, and $\mathbf{x}^{(i)}$ are the vectors associated to C_i. $s_b(\mathbf{u})$ measures the separation between the individual class means respect to the global mean face, and $s_w(\mathbf{u})$ measures the separation between vectors of each class respect to their own class mean. Alternatively we can define the scatter matrices as:

$$\mathbf{S}_b = \sum_{i=1}^{NC} \mathrm{P}(C_i)\left(\mathbf{m}^{(i)} - \mathbf{m}\right)\left(\mathbf{m}^{(i)} - \mathbf{m}\right)^{\mathrm{T}} \tag{6}$$

$$\mathbf{S}_w = \sum_{i=1}^{NC} \mathrm{P}(C_i)\mathrm{E}\left[\left(\mathbf{x}^{(i)} - \mathbf{m}^{(i)}\right)\left(\mathbf{x}^{(i)} - \mathbf{m}^{(i)}\right)^{\mathrm{T}}\right] \tag{7}$$

Then:

$$\gamma(\mathbf{u}) = \frac{\mathbf{u}^{\mathrm{T}}\mathbf{S}_b\mathbf{u}}{\mathbf{u}^{\mathrm{T}}\mathbf{S}_w\mathbf{u}} \ . \tag{8}$$

At this point is not difficult to demonstrate that the solution of our maximization problem is the solution of the generalized eigensystem:

$$\mathbf{S}_b\mathbf{w}^k = \lambda_k\mathbf{S}_w\mathbf{w}^k \tag{9}$$

$\mathbf{w}^k$ would be the fisherfaces and λ_k are the successive γ parameters associated with each fisherface. The system given by (9) is conventionally solved

by writing $\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{w}^k = \lambda_k \mathbf{w}^k$. Notice that this eigensystem does not have orthogonal eigenvectors because $\mathbf{S}_w^{-1}\mathbf{S}_b$ is not symmetric in general, then the fisherfaces would not be an orthogonal projection set. Another implementation problems are: the matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ are too big, and also $\mathbf{S}_w$ could be singular and then non-invertible. An easy way to solve these two problems, is to use a PCA decomposition previous to the FLD procedure. Then, the size of the scatter matrices would be small enough and depending on the criterion for dimensional reduction used, $\mathbf{S}_w$ will became non-singular. In this case, the eigensystem (9) will give reduced eigenvectors $\mathbf{v}^k$, that need to be transformed into the true eigenvectors $\mathbf{w}^k$ using $\mathbf{W}_{FF} = \mathbf{V}\,\mathbf{W}_{EF}$, where $\mathbf{W}_{EF}$ and $\mathbf{W}_{FF}$ are the PCA and Fisher projection matrices, respectively. With two classes we need only one γ to describe de between/within class scatter, then the projection space might have only one dimension. In the general case with *NC* classes, we need *NC-1* values of γ, and this corresponds to the dimension of the projection space. Given that (6) is the sum of *NC* matrices of rank one or less, *NC-1* of them are independent, and this gives *NC-1* possible values of non-zeros γ in (8) [6]. Figure 4 shows the complete FLD procedure. Thereafter we could adjust a criterion for the Fisher-space dimensional reduction. In analogy with PCA a good criterion would be to choose only *m* components ($m \leq NC\text{-}1$) obtained by the normalized *Residual Fisher Parameter*:

$$\mathrm{RFP}(m) = \sum_{k=m+1}^{NC-1} \lambda_k \Big/ \sum_{k=1}^{NC-1} \lambda_k \qquad (10)$$

The computational procedure to solve the eigensystem (9) is not as easy as PCA because of the non-symmetry of $\mathbf{S}_w^{-1}\mathbf{S}_b$. Solving the eigensystem for $\mathbf{S}_w^{-1}\mathbf{S}_b$ could be unstable, then there are other possibilities to solve this eigensystem using the fact that both matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ are symmetric. One of these procedures, described in [3], solve (9) by a sequence of symmetric eigensystems. Even if we does not need to invert $\mathbf{S}_w$ directly, the numerical algorithm must give the eigenvalues γ given by (8), then if $\mathbf{S}_w$ is singular the whole eigensystem became singular.

Considering *m* given by RFP(*m*) < 10^{-5} % will be good for standard applications. The advantage of FLD against PCA is that the information kept in the dimensional reduction is better for recognition purposes. However, there exists some drawbacks because FLD uses the particular class information and then is recommended to have a lot of images per class in the training, or at least a good characterization of each one. In other words, in PCA the convergence of the $\mathbf{R}$ estimator depends mostly on the total number of target images *N*, but in FLD the convergence of the scatter matrices estimators depends also on the numbers of target images per class. Then, the main drawback of FLD is that it could be over-adjusted on the target images, and then the recognition system may have an important lack of generalization that may be notice in the resulting system recognition rate.

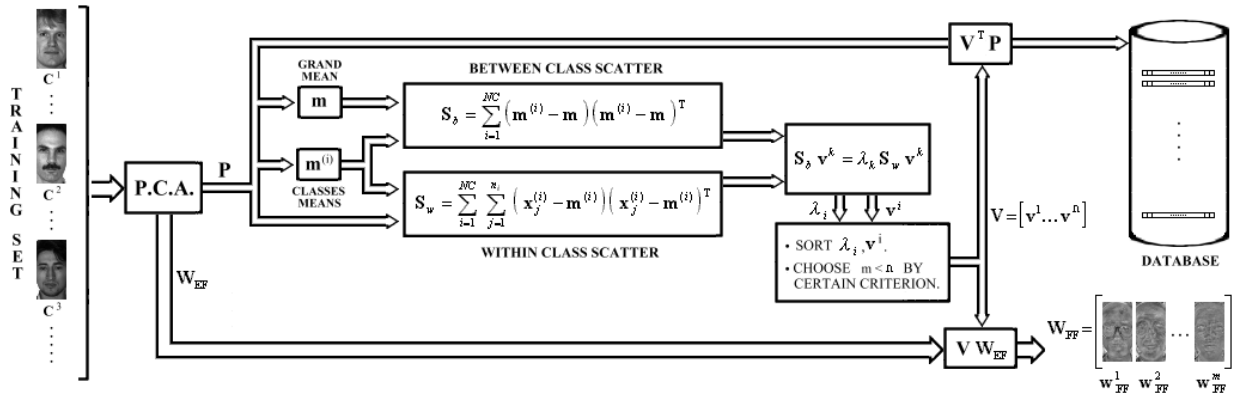

Fig. 4. FLD procedure used to obtain the projection matrix and the face projections vectors of the database. $\mathbf{P}$ is the matrix of PCA-projected vectors, $\mathbf{W}_{EF}$ is the PCA-projection matrix, and $\mathbf{W}_{FF}$ is the resulting Fisher projection matrix.

2.4 Evolutionary Pursuit - EP

The eigenspace-based adaptive approach proposed in [2], searches for the best set of projection axes in order to maximize a fitness function that measures at the same time the classification accuracy and generalization ability of the system (contrasting abilities in PCA and FLD projection methods). Because the dimension of the solution-space of this problem is too large, it is solved using a specific kind of Genetic Algorithm called *Evolutionary Pursuit* (EP). In this work, we are going to call the resulting projection axes (image vectors) EP-faces.

In order to obtain the EP-faces first we do an initial dimensional reduction using PCA, and then we apply the Whitening Transformation $\mathbf{\Gamma}^{-1/2}$ (equivalent to a Mahalanobis metric system, see 2.3), which is used in the system in order to obtain a non-orthogonal set of projection axes due to the effect of stretches and shrinks of axes. In the Whitened-PCA space we do several rotations between pair of axes and then a subset of the rotated Whitened-PCA axes are selected to maximize the fitness function. These rotations are coded using a chromosome representation. The chromosome structure, shown in Figure 5, use 10 bit for each rotation angle α_k (between 0 and $\pi/2$), and a number of bits a_i equal to the number of Whitened-PCA components (*m*), in order to select the subset of axes to be rotated. Rotation and selection of axes are represented by a single matrix $\mathbf{Q}$. Notice that the number of possible rotations between axes would be $m(m-1)/2$, so the number of bits for each chromosome is $5\,m(m-1)+m$, and the size of the genome-space (too large to search it exhaustively) is $2^{5\,m(m-1)+m}$.

Each chromosome represents a certain projection system. In order to evaluate this projection the following fitness function is used:

$$\zeta(\alpha_k, a_i) = \zeta_a(\alpha_k, a_i) + \lambda\,\zeta_s(\alpha_k, a_i) \qquad (11)$$

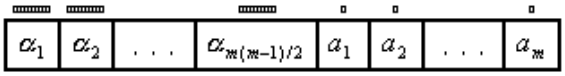

Fig. 5. Chromosome structure representing a given solution of EP-faces.

$\zeta_a(\alpha_k, a_i)$ measures the accuracy of the system, $\zeta_s(\alpha_k, a_i)$ measures its generalization ability, and λ is a positive constant that determines the importance of the second term against the first one. The generalization ability is computed as (remember $\mathbf{S}_b$ given by (6)):

$$\zeta_s(\alpha_k, a_i) = \sqrt{\sum_{i=1}^{NC} \left(\mathbf{m}^{(i)} - \mathbf{m}\right)^T \left(\mathbf{m}^{(i)} - \mathbf{m}\right)} \qquad (12)$$

where *NC* is the number of classes, $\mathbf{m}$ is the global mean, and $\mathbf{m}^{(i)}$ is the mean of the corresponding class C_i. In this way $\zeta_s(\alpha_k, a_i)$ measures the generalization ability in the same way of PCA and Fisher (using separation between classes). The accuracy measure $\zeta_a(\alpha_k, a_i)$ proposed in [2] is just the system recognition rate over the target images using the top choice (closed universe). In our implementation we realize that this measure became always 100% when the number of classes is smaller than the dimension of the reduced space (we use only 15 classes against the 369 used in [2]). Then we use the top 2 match identity for that.

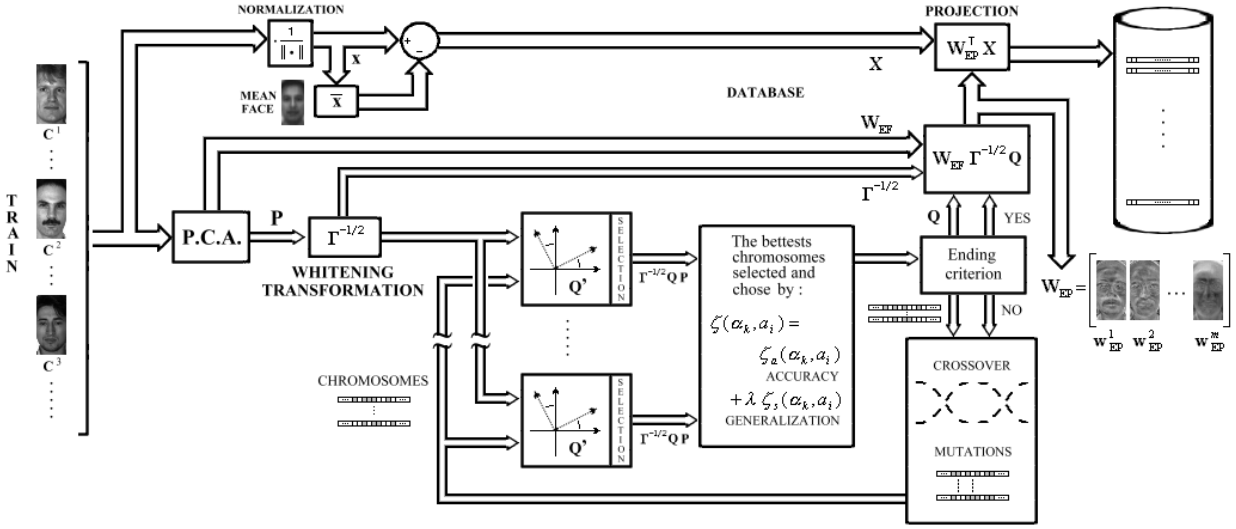

Fig. 6. EP procedure used to obtain the projection matrix and the face projections vectors of the database. $\mathbf{P}$ is the matrix of PCA-projected vectors, $\mathbf{W}_{EF}$ is the PCA-projection matrix, $\mathbf{Q}$ is the selected rotation-reduction matrix, and $\mathbf{W}_{EP}$ is the resulting EP projection matrix.

As usual, in order to find the maximal value of the fitness function, we start with a random set of chromosomes and we search for the best one using the GA iteration procedure. The operators used to create a new set of chromosomes per iteration are: proportionate selection, pre-selection of parents in proportion to their relative fitness; two-point crossover; and fixed probability of mutation (each bit of the chromosome is given a fixed probability of flipping). The whole training process is represented in figure 6.

2.5 Similarity Matching Methods

2.5.1 Euclidean Distance

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{p} \left(x_i - y_i\right)^2} \ . \tag{13}$$

From a geometrical point of view this distance measures the difference between the reduced face vectors and the difference between the reconstruction error of each vector, given by the difference of the norms.

2.5.2 Cosine Distance

$$\cos(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^{\mathrm{T}}\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} \ . \tag{14}$$

Given that the image vectors are normalized and located over an hyper-sphere surface, the angle between them represents the distance above this surface. Although in the reduced space the norm of the vectors decrease depending on the reconstruction error. This distance does not take into account the reconstruction error of each vector.

2.5.3 Mahalanobis Distance

$$d(\mathbf{x},\mathbf{y}) = \left(\mathbf{x}-\mathbf{y}\right)^{\mathrm{T}} \ \mathbf{R}^{-1} \ \left(\mathbf{x}-\mathbf{y}\right), \tag{15}$$

where $\mathbf{R}$ is de correlation matrix of face images.

From a geometrical point of view this distance, as a different metric system, has a scaling effect in the image space. Taking into consideration the face image subset, directions in which a greater variance exist are compressed and directions in which a smaller variance exist are expanded.

In (15) we suppose $\mathbf{x}$ and $\mathbf{y}$ are vectors in the image space. Notice that given $\mathbf{R}^{-1} = \mathbf{W}_{\mathrm{EF}}^{\mathrm{T}} \Gamma^{-1} \mathbf{W}_{\mathrm{EF}}$, then:

$$d(\mathbf{x},\mathbf{y}) = \left\{\mathbf{W}_{\mathrm{EF}} \Gamma^{-1/2} \left(\mathbf{x}-\mathbf{y}\right)\right\}^{\mathrm{T}} \left\{\mathbf{W}_{\mathrm{EF}} \Gamma^{-1/2} \left(\mathbf{x}-\mathbf{y}\right)\right\}. \tag{16}$$

That means in the PCA space the Mahalanobis distance is equivalent to the Euclidean distance, weighting each component by the inverse correspondent projection variance (square rooted eigenvalue). In general this happened when the different components are uncorrelated between them. Another way to do this normalization is using the standard PCA and after that changing the eigenface matrix by $\mathbf{W}_{EF}\Gamma^{-1}$. This is called *Whitening Transformation* of PCA [2] (see 2.2.3). Then using the Euclidean distance over the whitened PCA vectors, we have the same effect as using the Mahalanobis distance.

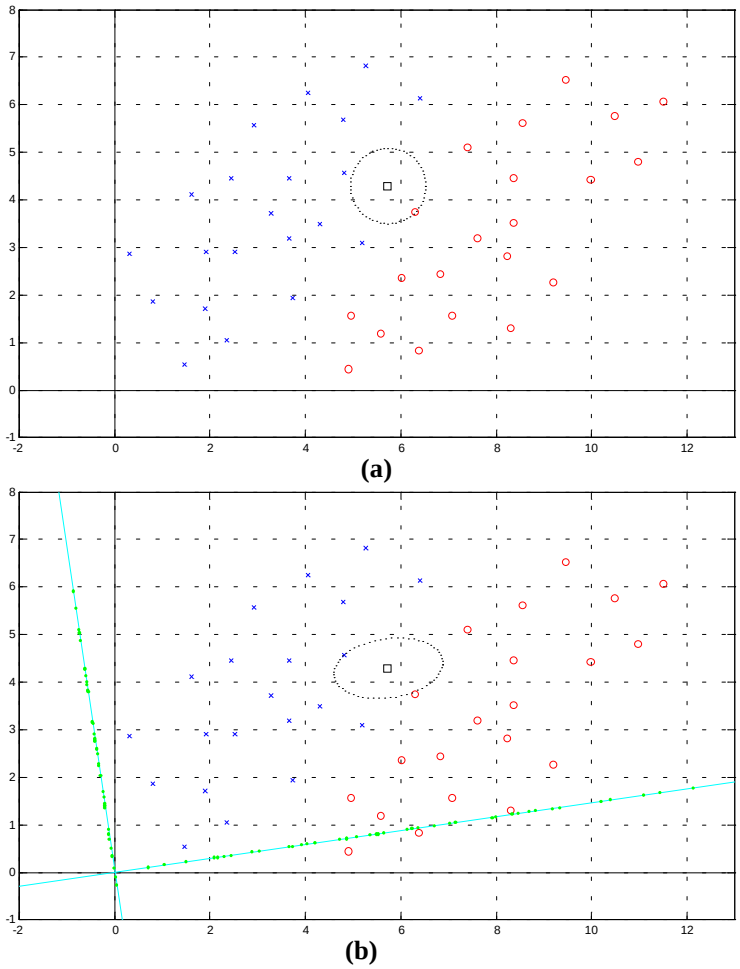

Fig. 7. **(a)** Between two normal distributed classes the Euclidean metric recognize a new point (square) as a circle-class. **(b)** In the same example the Mahalanobis metric, weighting the PCA axes by the inverse projection variances, recognize the new point as a cross-class (Bayes optimal).

The Mahalanobis metric could be better than the Euclidean metric for the recognition matching problem if each class is normal distributed. Figure 7a and 7b shows how the same problem gives different results under Euclidean or

Mahalanobis decision rules. In this figures there are two 2D normal distributed classes (crosses and circles) and we must recognize the class of a new point (the square). In this example the sets of normal distributed points are generated knowing the real probability densities, so we know that the best choice for the new point, using the Bayes criterion, is the cross class. In Figure 7a we use the Euclidean minimum-distance criterion that identify the new point as a circle. In Figure 7b however, the Mahalanobis minimum-distance criterion identify the new point as a cross. The last figure also shows the scale effect of the Mahalanobis metric in the PCA projection axes, because the original circle is warped into an ellipse accordingly to the projection variance in each PCA axe.

2.5.4 SOM Clustering

Self-Organizing Maps (SOM) are used as associative networks to match the projected query face with the corresponding projected database face. The use of a SOM to implement this module improves the generalization ability of the system.

The SOM approach uses reference vectors $\mathbf{m}_i$, the so-called SOM codewords, to approximate the probability distribution of the faces in a 2D map. In the training phase of the SOM a clustering of the reduced vector faces is carried out. Thereafter the SOM is transformed in an associative network by a labeling of all its nodes.

There are two alternatives to implement this module: the standard SOM algorithm and the Dot-Product SOM (DP-SOM) [8]. Using either of these two approaches to find the reference vectors, the following step is to find the best matching of each target image and to see in which place of the map it is going to be located. After that a so-called *labeling* algorithm allows to associate a fixed face identity for each node in the map.

2.5.5 Fuzzy Similarity Matching

$$S(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m} \min\{\mu_i(\mathbf{x}),\mu_i(\mathbf{y})\} - \alpha \sum_{i=1}^{m} \max\{\mu_i(\mathbf{x}) - \mu_i(\mathbf{y}),0\} - \beta \sum_{i=1}^{m} \max\{\mu_i(\mathbf{y}) - \mu_i(\mathbf{x}),0\} \quad (17)$$

where $\mu_i(\mathbf{x})$ is a membership function associated with the i-component of vector $\mathbf{x} \in R^m$. This similarity measure, originally proposed in [7], is a fuzzy implementation of the Feature Contrast model from Tversky [1]. The first sum measures the common features (intersection) and the other two represent the distinctive features (difference in the two possible ways). The positive parameters α and β allow to adjust the contrast of the three kind of features. By choosing $\alpha \neq \beta$ it is possible to introduce asymmetries between the subject-referent comparison. This model considers that all the features are independent, and that is the case of PCA and WPCA, but not in FLD and EP. In our implementation we normalize each feature of PCA (in WPCA it is not necessary) and we choose $\mu_{i(\mathbf{x})}$ linear between –1 and 1, with each component x_i normalized (null mean and unitary variance).

3 Comparisons among the approaches

In order to compare the described methods we made several simulations using the Yale University - Face Image Database [9], which corresponds to a database with a small number of classes (15). In http://www.cec.uchile.cl/~jruizd/faces are shown some of the face images used for training, some examples of images obtained as the projection axes for the several tested procedures (Eigenfaces, Fisherfaces and EP-faces), and also a PCA reconstruction example.

For testing purposes we use 150 images of 15 different classes (only 10 of the 11 images per class of the Yale Database were considered) in order to form the target and query sets. We preprocessed the images by masking them in windows of 100 x 200 pixels placing the several face features in the same relative places.

Table 1. Mean recognition rates using different numbers of training images per class, and taking the average of 20 different training sets. The small numbers are the standard deviation of each recognition rate. The EP-Whitening case is omitted because EP already consider this transformation.

	im./class	**axes**	Euclidean	cos(·)	SOM	FFC	Whitening Euclidean	Whitening cos(·)	Whitening SOM	Whitening FFC
PCA	**6**	56	87,9 6,2	86,0 6,8	84,6 7,0	77,1 10,1	64,7 9,4	79,3 11,6	64,7 10,5	77,1 10,1
FISHER		14	91,5 6,6	91,6 6,5	90,3 6,7	83,9 9,3	91,9 5,8	92,6 5,6	92,1 6,2	85,6 8,3
E.P.		16	81,2 9,0	85,3 8,7	83,7 9,8	77,2 8,0	-	-	-	-
PCA	**5**	34	88,7 3,8	87,1 4,1	86,0 5,1	78,5 8,1	69,5 8,9	83,2 9,0	66,1 10,5	78,5 8,1
FISHER		14	92,2 5,7	91,7 6,2	90,3 6,4	85,1 9,1	92,3 4,7	92,4 5,7	92,1 5,3	85,4 8,5
E.P.		13	84,1 5,7	87,7 6,6	86,7 7,6	78,7 6,8	-	-	-	-
PCA	**4**	46	87,3 3,9	86,7 3,9	84,8 3,6	77,6 5,2	72,9 5,5	84,4 5,6	66,7 6,5	77,6 5,2
FISHER		14	90,3 4,5	91,1 5,0	90,3 4,4	84,4 5,9	90,4 4,2	91,0 4,4	90,1 4,7	82,9 5,7
E.P.		18	83,6 4,6	86,9 4,7	85,0 5,0	74,7 6,0	-	-	-	-
PCA	**3**	35	86,6 4,0	85,4 3,9	82,0 5,6	77,9 4,6	75,0 5,6	84,8 5,4	67,4 6,9	77,9 4,6
FISHER		14	89,0 3,6	90,4 4,0	87,4 4,0	80,7 6,3	88,9 3,1	89,9 3,9	88,7 3,9	81,5 3,4
E.P.		14	81,1 4,3	86,9 3,7	82,5 3,7	75,9 4,4	-	-	-	-
PCA	**2**	26	82,7 5,9	80,8 5,9	76,2 7,9	71,1 5,9	75,6 4,9	82,1 4,6	60,8 7,3	71,1 5,9
FISHER		14	81,5 5,6	82,2 5,8	79,4 5,8	69,3 8,6	80,7 4,7	82,8 4,9	78,8 5,8	73,6 6,2
E.P.		14	77,8 5,6	81,2 5,3	76,0 7,3	70,0 7,4	-	-	-	

In Table 1 we show the results of several simulations using different kind of representations and similarity matching methods. The criterion used for the PCA representation is RMSE<1.5%. In the Fisher representation we always use RFP<10^{-5}% but in the PCA starting space we change the RMSE criterion depending on the numbers of target images used until the $\mathbf{S}_w$ matrix became non-singular. In the EP representation we always started with RMSE<7% to improve

the initial dimensional reduction. Notice that we never fix the number of axes used in each representation, and thereafter this is always a result of the training process. Figure 8 shows the variations of each system recognition rate using different number of training images per classes.

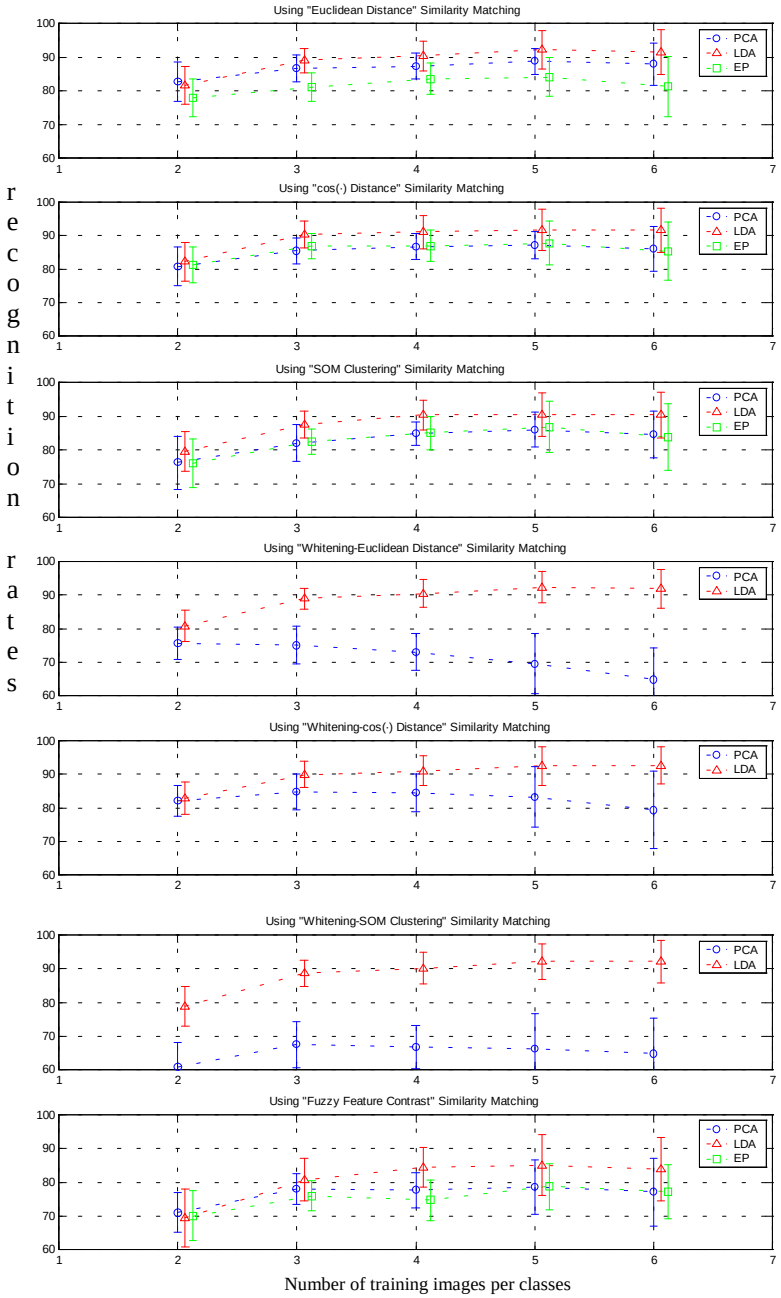

Fig. 8. Graphics of the mean recognition rates using different numbers of target images per class, and taking the average of 20 different target sets.

For each simulation we used a fixed number of target images, choosing the same types of images per class (in the Yale Face Database terminology the 10 images per class are: center-light, w/glasses, happy, left-light, w/no glasses, right-light, sad, sleepy, surprised, and wink). We take the average of results for 20 different sets of images for each fixed number of target images. All the images not included in the target set are used in the query set. Also because the number of axes are always a result of training, the number of axes shown in Table 1 are the round average of several simulations.

In our results we can notice that the number of axes selected are almost always equal or larger than the number of classes (15), except in the Fisher case. We can also see that the best algorithms always are obtained with the Fisher representation, and the difference against the other representations decreases when the number of target images per class decreases, showing that the FLD discrimination ability strongly depends on the number of target images per class. The best results are almost always obtained with FLD-cosine, and the only exception was using 2 target images per class when the Withening-FLD-cosine wins. The systems that seem to be as efficient as FLD-cosine are SOM, and Withening-FLD-cosine.

In the best results we obtained using FFC, we used an asymmetric subject-referent comparison: α=0.5 β=5. This means that in the question "how similar are the subject faces to the referent face?" the answer focus more on the features of the referent (the unknown face). The SOM system considerably decreases its recognition rate as the number of target images decrease. This probably happened because we always use a SOM-map of 20x20 nodes, and when the number of target images is small, the recognition ability depends mostly on the labeling procedure, which became very unpredictable. The Withening-FLD-cosine system has maintained its recognition ability more than the other systems when the number of target images per class decreases.

We can also see that the EP-systems always performed worse than the FLD systems. At the same time, we realized that the number of axes selected are always of the same order of the number of classes, then the accuracy pursuit (mostly dependent on the top 2 match) seems to fail for this reason, and then FLD kept the advantage. The worst results seems to be the obtained with Whitening-PCA-Euclidean and Whitening-PCA-SOM, and against the result of the cosine-based systems we can see that the changes in the norms of the vectors seems to confuse the recognition ability.

4 Conclusions and System Outline

We made an extensive analysis of different Eigenspace-based approaches, separating the representation problem from the similarity matching method employed, and using a database with a small number of classes. We also use the Whitening Transformation as a Mahalanobis metric system before the initial PCA processing, in order to match PCA and FLD against EP. We saw important differences between the recognition rated reached using Euclidean and cosine similarity matching methods. Without the whitening processing we confirm that this is due the consideration of reconstruction error on the similarity measure,

because the difference between them became appreciable when the number of projection axes decrease. Using whitening processing the difference considerably increase because the vectors' norm changes due the scaling effect of this procedure. In order to obtain the best recognition rates we saw that the cosine similarity matching works better, and that the whitening-cosine based methods seem to keep their recognition ability with fewer target images. Concerning the lower recognition ability of our EP implementation, we conclude that this is not the best representation method for a system with small number of classes (only 15!), or at least the accuracy measure is not appropriate. The recognition ability of the FFC implementation is also restricted due to the small number of classes.

An important goal of this paper was to change the standard block of similarity matching for a SOM. Notice that in this way the structure of the identification system (see figure 1) changes because we do not need the database reduced representation anymore, because now this information would be appropriately included in the SOM reference vectors (codewords). In our simulations we realize that this kind of identification system works as good as the standard ones, concerning the recognition rates. Nevertheless, this recognition rate is limited because the small number of classes considered. But an even more interesting feature of this approach is the possibility to adapt itself to changes in the faces. This has a direct application in adaptive security access systems where the persons to be recognized would be constantly viewed by the system. Specifically when a person arrive, the neural system will carried out the recognition, and after that the SOM will perform one training iteration. This iteration might be done with a small learning rate and modifying only the nearest neighbors of the winner neuron. The labeling procedure will be re-applied. In this way the SOM map will adapt itself to the changes of faces like the change due the bear, hair, or even age evolution. In this sense this system represents a robust and adaptive control access identification system.

As future work we want to perform our comparative study on a larger database, like FERET. Using such a database with a large number of classes, we believe that the positive properties of EP, FFC and SOM will be noticed.

Acknowledgement

This research was supported by the DID (U. de Chile) under Project ENL-2001/11 and by the join "Program of Scientific Cooperation" of CONICYT (Chile) and BMBF (Germany).

References

1. A. Tversky, "Features of Similarity", *Psychological Review*, vol. 84, no. 4, pp. 327-352, July 1977.
2. C. Liu and H. Wechsler, "Evolutionary Pursuit and Its Application to Face Recognition", *IEEE Trans. Patt. Analysis and Machine Intell.*, vol. 22, no. 6, 570-582, 2000.

3. D.L. Swets and J.J. Weng, “Using Discriminant Eigenfeatures for Image Retrieval”, *IEEE Trans. Patt. Analysis and Machine Intell.*, vol. 18, no. 8, 831-836, 1996.
4. M. Golfarelli, D. Maio and D. Maltoni, “On the Error-Reject Trade-Off in Biometric Verification Systems”, *IEEE Trans. Patt. Analysis and Machine Intell.*, vol. 19, no. 7, 786-796, 1997.
5. M. Turk, and A. Pentland, “Eigenfaces for Recognition”, *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
6. R.O. Duda, P.E. Hart, and D.G. Stork, “Pattern Classification”, Chap. 3, Second Edition, 2001.
7. S. Santini, and R. Jain, “Similarity Measures”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 871-883, September 1999.
8. T. Kohonen, “Self-Organized Maps”, Springer, 1997.
9. Yale University Face Image Database, publically available for non-commercial use, http://cvc.yale.edu/projects/yalefaces/yalefaces.html