Eigenspace-based Recognition of Faces: Comparisons and a new Approach

Pablo Navarrete and Javier Ruiz-del-Solar
Dept. of Electrical Engineering, Universidad de Chile
Email: {pnavarre, jruizd}@cec.uchile.cl

Abstract. 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. A first goal of this paper is to present a comparison between some of these different approaches. A second goal is to outline an adaptive, neural-based Security Access Control System.

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. 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. A first goal of this paper is to present a comparison between some of these different approaches. The comparison considers the use of three different projection methods (Principal Component Analysis, Fisher Linear Discriminant and Evolutionary Pursuit) and four different similarity matching criteria (Euclidean-, Cosines- and Mahalanobis-distance and Self-Organizing Map). A second goal of the paper is to outline an adaptive, neural-based Security Access Control System, whose main characteristic is the adaptive learning of changes in the face of the database persons.

The article is structured as follows. Different eigenspace-approaches are described in section 2. In section 3 these different approaches are compared. Finally, in section 4 a neural-based Security Access Control System is outlined and 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 feature vectors) has a lower dimension than the image space (given by the number of pixels in the 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 image. In the off-line phase are also obtained the reduced representations of each database images. These representations are the ones to be used in the recognition process.

2.1. General Approach

Figure 1 shows the block diagram of a generic eigenspace-based face recognition system. A preprocessing module performs a normalization to transform the face image into a unitary vector and then a subtraction of a so-called *mean face* ($\bar{\mathbf{x}}$). Normalization is necessary to initialize each face vector $\mathbf{x}$ with the same energy. The mean face is previously obtained in the off-line training phase.

After that, $\mathbf{x}$ 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, starting with vectors $\mathbf{x}$ in R^N (with N the vector image dimension), and obtaining projected vectors $\mathbf{q}$ 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 $\mathbf{x}$ will be possible. For face recognition tasks is not critical the reconstruction ability of the projection.

The *Similarity Matching* module compares the similarity of the reduced representation of the query vector face $\mathbf{q}$ 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 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 [3].

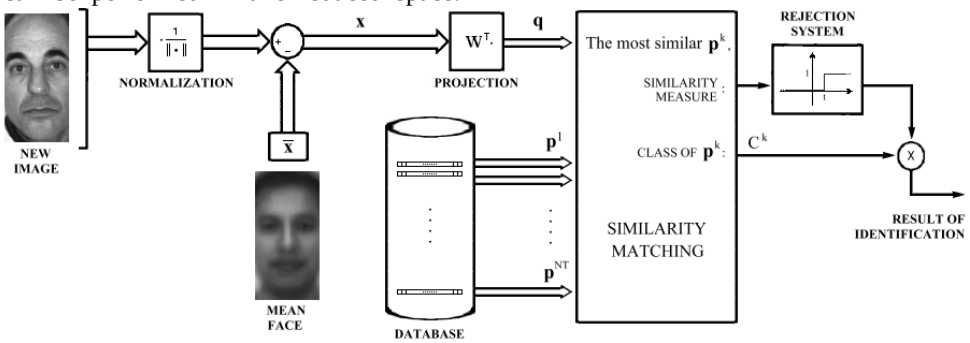

Figure 1. Block diagram of the recognition system based on Eigenspace projection.

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 $\mathbf{R} \in R^{N \times N}$:

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

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 eigenface. 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. We have a database of NT face images (the training set), and then we need to estimate the correlation matrix just by taking the corresponding averages in the training 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 training 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 training images all the variance must be projected into the hyperplane subtended by the training 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 training 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 $\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{X}\,\mathbf{v}^k = \lambda_k\,\mathbf{X}\mathbf{v}^k$ represent both systems at the same time.

Now we can solve the reduced eigensystem of $\mathbf{X}^{\mathrm{T}}\mathbf{X} \in R^{NT \times NT}$. The correspondent eigenvalues are just the eigenvalues of the original system, and the eigenfaces are represented by $\mathbf{w}^k = \mathbf{X}\mathbf{v}^k$, and to be normalized they must be divided by $\sqrt{\lambda_k}$. 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* [2]:

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

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

2.2.2. 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. In order to define the mathematical structure under FLD, first we define the parameter $\gamma(\mathbf{u})$ 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 $s_b(\mathbf{u})$ and $s_w(\mathbf{u})$ given by:

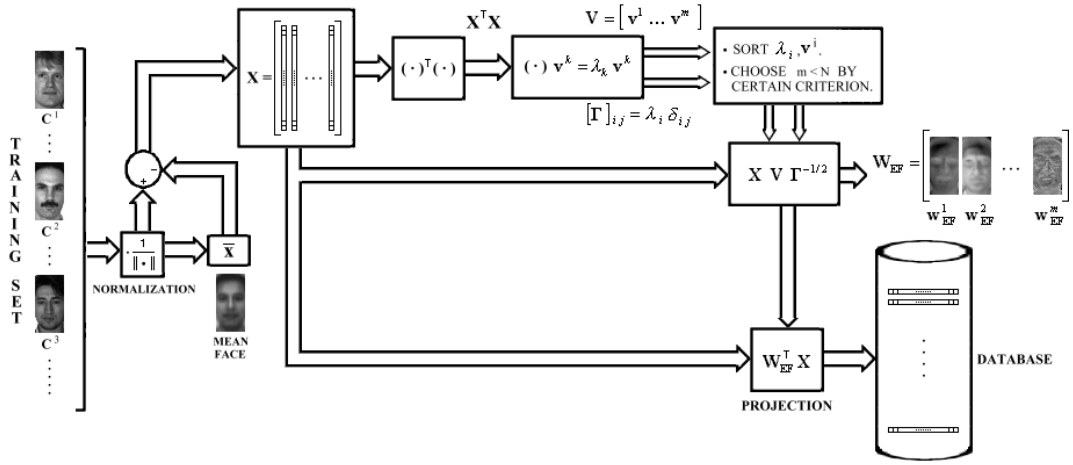

Figure 2. PCA procedure used to obtain the projection matrix and the face projections vectors of the database.

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

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

where $\mathbf{m}$ 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 $s_w(\mathbf{u})$ measure the separation between vectors of each class respect to their own class mean. Alternative we define the scatter matrices:

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

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

then:

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

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

$$\mathbf{S}_b\mathbf{w}^k = \lambda_k\mathbf{S}_w\mathbf{w}^k \quad (9)$$

Then $\mathbf{w}^k$ would be the fisherfaces and λ_k are the successive γ parameters associated with each fisherface. Notice that the eigensystem of $\mathbf{S}_w^{-1}\mathbf{S}_b$ does not have orthogonal eigenvectors because this matrix 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 at the same time is to use PCA previous to the FLD procedure. Then, the size of the scatter matrices would be large enough and depending on the criterion for dimensional reduction $\mathbf{S}_w$ will became non-singular. Figure 3 shows the 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 obtained by the normalized *Residual Fisher Parameter*:

$$\mathrm{RFP}(m) = \sum_{k=m+1}^{N}\lambda_k \Big/ \sum_{k=1}^{N}\lambda_k \quad (10)$$

Considering *m* given by $\mathrm{RFP}(m) < 10^{-5}$ % 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. Although there exists some drawbacks, because FLD use 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 training images *N*, but in FLD the convergence of the scatter matrices estimators depends also on the numbers of images per class. Then, the main drawback of FLD is that could be over-adjusted on the training images, and then the recognition system may have an important lack of generalization that may be notice in the recognition rate.

2.2.3. Evolutionary Pursuit - EP

The eigenspace-based adaptive approach proposed in [1], searches for the best set of projection axes in order to maximize a fitness function, measuring at the same time the classification accuracy and generalization ability of the system. 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 the proposed representation, we are going to call the projection axes (image vectors) as EP-faces.

In order to obtain the EP-faces first we do an initial dimensional reduction using PCA, and then we use the Whitening Transformation (equivalent to a Mahalanobis metric system, see 2.3). In the Whitened-PCA space we do several rotations between pair of axes and then we select a subset of the rotated Whithened-PCA axes. This

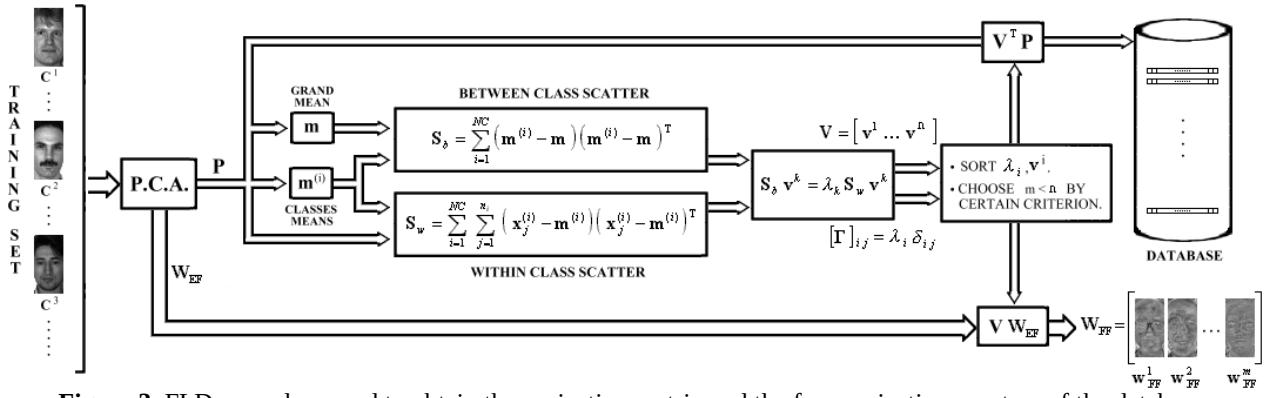

Figure 3. FLD procedure used to obtain the projection matrix and the face projections vectors of the database.

transformation is coded using a chromosome representation, the chromosome structure use 10 bit for each angle α_k (between 0 and $\pi/2$), and a number of bits a_i equal to the number of Whithened-PCA components (m), in order to select a subset of axes. Notice that the number of possible rotations between axes would be $m(m-1)/2$, so the number of bits for each chromosome is $5m(m-1)+m$, and the size of the genome-space (too large to search it exhaustively) is $2^{5m(m-1)+m}$.

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

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

$\zeta_a(\alpha_k, a_i)$ measures the accuracy, $\zeta_s(\alpha_k, a_i)$ measures the 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:

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

where $\mathbf{m}$ is the global mean and $\mathbf{m}^{(i)}$ is the mean of the corresponding class C_i. The accuracy measure $\zeta_a(\alpha_k, a_i)$ proposed in [1] is just the recognition rate of training face images as the top choice. But in our implementation we realize that this measure became always 100% when the number of classes is small respect to the dimension of the reduced space (we use only 15 classes against the 369 used in [1]). Then we use the top 2 identity for that.

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 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, exchange the selection between the crossover points; and fixed probability mutation, each bit of the chromosome is given a fixed probability of flipping. The whole training process is represented in figure 4.

2.3. Similarity Matching Methods

2.3.1. Euclidean Distance

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

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 norm of each one.

2.3.2. Cosine Distance

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

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.

2.3.3. Mahalanobis Distance

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

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.

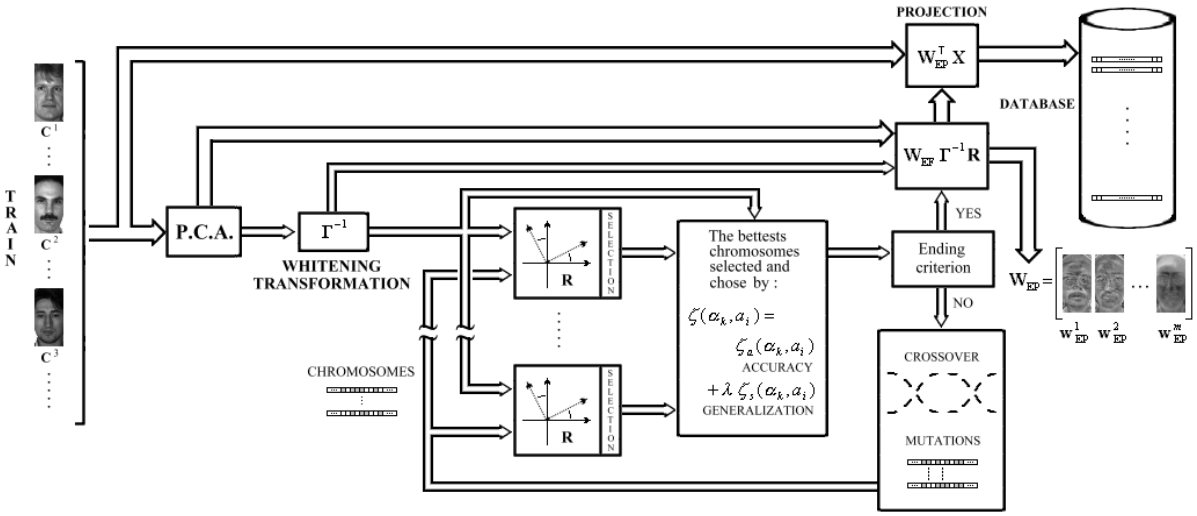

Figure 4. EP procedure used to obtain the projection matrix and the face projections vectors of the database.

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

$$d(\mathbf{x},\mathbf{y}) = \{\mathbf{W}_{EF}(\mathbf{x}-\mathbf{y})\}^{T} \Gamma^{-1} \{\mathbf{W}_{EF}(\mathbf{x}-\mathbf{y})\}. \quad (16)$$

So, in the PCA space the Mahalanobis distance is equivalent to the Euclidean distance, weighting each component by the inverse correspondent eigenvalues (or projection variance). 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 [1]. Then using the Euclidean distance over the whitened PCA vectors, we have the same effect as using the Mahalanobis distance.

2.3.4. SOM Clustering

In 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 his nodes.

There are two alternatives to implement this module: the standard SOM algorithm and the Dot-Product SOM (DP-SOM) [4]. Using either of these two approaches to find the reference vectors, the following step is to find the best matching of each training image and to see in which place of the map they are going to be located. After that a so-called *labeling* algorithm allows to associate a fixed face identity for each node in the map.

3. Comparison among the approaches

In order to test the described methods we made several simulations based in the Yale University - Face Image Database, which corresponds to a database with a low number of classes (15). We use 150 images of 15 different classes (only 10 of the 11 images per class where considered). First we preprocessed the images by masking them in windows of 100 x 200 pixels placing the several face features in the same relative places.

In figure 5 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 training 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.

For each simulation we used a fixed number of training images, using the same types of images per class (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 20 different set of images for each fixed number of training images. All the images not used for training are used for testing. Also because the number of axes are always a result of training, the number of axes shown in figure 5 are the round average of several simulations.

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

The SOM and DP-SOM systems considerably decrease their recognition rate as the number of training images decrease. This probably happened because we always use a map of 20x20 nodes, and when the training images are few, 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 training 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) seems to fail for this reason, and then FLD kept the advantage. The worst results seems to be the obtained with Whithening-PCA-Euclidean and Whithening-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 the recognition capabilities of different Eigenspace-based approaches, separating the representation problem from the similarity matching method employed and using a database with a low 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 seems to keep their recognition ability with fewer training images. Concerning the lower recognition ability of our EP implementation, we conclude that this is not the best representation method for a small number of classes, or at least the accuracy measure is not appropriate.

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. But an interesting feature of this approach is the possibility to adapt itself to the changes of 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 new 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 value of α and modifying only the nearest neighbors. 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 represent a robust control access identification system.

As future work we want to perform our comparative study on a larger training database, like FERET. We want also to implement the here-outlined adaptive security-access control system.

Acknowledgements

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] 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.

[2] 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.

[3] 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.

[4] T. Kohonen, "Self-Organized Maps", Springer, 1997.

	im./class	**axes**	Euclidean	cos(·)	DP-SOM	SOM	Whitening Euclidean	Whitening cos(·)	Whitening DP-SOM	Whitening SOM
PCA RMSE < 1.5%	**6**	56	87.9 6.2	86.0 6.8	84.2 6.3	84.6 7.0	64.7 9.4	79.3 11.6	73.4 10.5	64.7 10.5
FISHER RMSE < 1.5%		17	91.5 6.6	91.6 6.5	89.3 7.4	90.3 6.7	91.9 5.8	92.6 5.6	88.9 6.2	92.1 6.2
E.P. RMSE < 7%		16	81.2 9.0	85.3 8.7	82.6 9.4	83.7 9.8	-	-	-	-
PCA RMSE < 1.5%	**5**	34	88.7 3.8	87.1 4.1	83.7 4.8	86.0 5.1	69.5 8.9	83.2 9.0	76.5 8.8	66.1 10.5
FISHER RMSE < 1.5%		15	92.2 5.7	91.7 6.2	86.7 7.2	90.3 6.4	92.3 4.7	92.4 5.7	89.3 6.6	92.1 5.3
E.P. RMSE < 7%		13	84.1 5.7	87.7 6.6	84.7 6.8	86.7 7.6	-	-	-	-
PCA RMSE < 1.5%	**4**	46	87.3 3.9	86.7 3.9	81.2 4.9	84.8 3.6	72.9 5.5	84.4 5.6	77.1 5.1	66.7 6.5
FISHER RMSE < 3.0%		18	90.3 4.5	91.1 5.0	85.1 5.5	90.3 4.4	90.4 4.2	91.0 4.4	86.6 5.7	90.1 4.7
E.P. RMSE < 7%		18	83.6 4.6	86.9 4.7	81.8 4.4	85.0 5.0	-	-	-	-
PCA RMSE < 1.5%	**3**	35	86.6 4.0	85.4 3.9	77.7 4.6	82.0 5.6	75.0 5.6	84.8 5.4	72.7 5.2	67.4 6.9
FISHER RMSE < 4.5%		15	89.0 3.6	90.4 4.0	81.7 5.6	87.4 4.0	88.9 3.1	89.9 3.9	83.0 4.1	88.7 3.9
E.P. RMSE < 7%		14	81.1 4.3	86.9 3.7	76.6 3.0	82.5 3.7	-	-	-	-
PCA RMSE < 1.5%	**2**	26	82.7 5.9	80.8 5.9	64.9 7.6	76.2 7.9	75.6 4.9	82.1 4.6	62.4 6.7	60.8 7.3
FISHER RMSE < 12.0%		15	81.5 5.6	82.2 5.8	66.8 9.2	79.4 5.8	80.7 4.7	82.8 4.9	69.0 9.1	78.8 5.8
E.P. RMSE < 7%		14	77.8 5.6	81.2 5.3	69.9 5.6	76.0 7.3	-	-	-	-

Figure 5. 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.