Data Dimensionality Reduction for Face Recognition

Georgy Kukharev, Paweł Forczmański
Technical University of Szczecin, Faculty of Computer Science and Information Systems,
Zolnierska Str. 49, 71-210 Szczecin
email: {gkoukharev,pforczmanski}@wi.ps.pl

Abstract. In the process of image recognition in most of the applications there is a problem of gathering, processing and storing large amounts of data. A possible solution for reducing this amount and speeding-up the computations is to use some sort of data reduction. Efficient reduction of stored data without losing any important part requires an adaptive method, which works without any supervision. In this article we discuss a few variants of a two-step approach which involves Karhunen-Loeve Transform (KLT) and Linear Discriminant Analysis (LDA). The KLT gives a good approximation of input data,however it requires a large number of eigenvalues. The second step reduces data dimensionality further using LDA. The efficiency of KLT depends on quality and quantity of input data. In the case, when only one image in a class is given as input, its features are not stable in comparison with other images in other classes. In the article we present a few methods of solving this problem, which is an improvement of the ideas presented in [6, 9].

Key words: dimensionality reduction, face recognition, eigenvectors, KLT, PCA, LDA

1. Introduction

In the practice of digital image recognition there is a widely known problem of huge dimensionality of input data. Each image which is to be recognized (no matter which technique we use) consists of thousands of pixels, where each pixel is represented by multi-byte value. Recognizing this kind of data involves time-consuming computational tasks. To solve the problem of high dimensionality many methods of reduction have been developed. They use both strictly deterministic and stochastic approaches. When we can describe the structure or can build a well represented set of input data, it is most suitable to use a deterministic approach, for example PCA - Principal Component Analysis [17], ICA - Independent Component Analysis [10, 16, 18], LDA - Linear Discriminant Analysis [12], HMM - Hidden Markov Models [5, 8, 11] and others [4]. On the other hand, if we can neither properly estimate the structure of data, nor collect representative elements of the whole set, we should use some sort of soft computing approach (artificial neural networks [2], rough sets, fuzzy logic, genetic algorithms [3] etc.) In this article, the problem of dimensionality reduction by means of deterministic approaches is described and appliedto practical tasks of human face recognition. For these tasks proper representation of input data can be easily provided.

2. Preparation of Data

The results of recognition strongly depend on the kind of features we use. Feature extraction can be carried out, based on one of three approaches: selection, reduction or selection combined with reduction. Proper description of a face with collected features is a very important subtask of the recognition problem. Some methods of feature selection are presented in Figure 1, where 0 – base face, 1 – down-scaled face, 2 – spectral representation, 3,4 – row or column concatenation, 5 – elastic model, 6 – characteristic points and geometric distances, 7 – combination of rows and columns. In our experiments, we used features obtained by row and column decomposition of the base face image. Dimensions of input dataset (DIM) are presented in table 1, where M and N describe input image's height and width and $m << M, n << N$ after scalling.

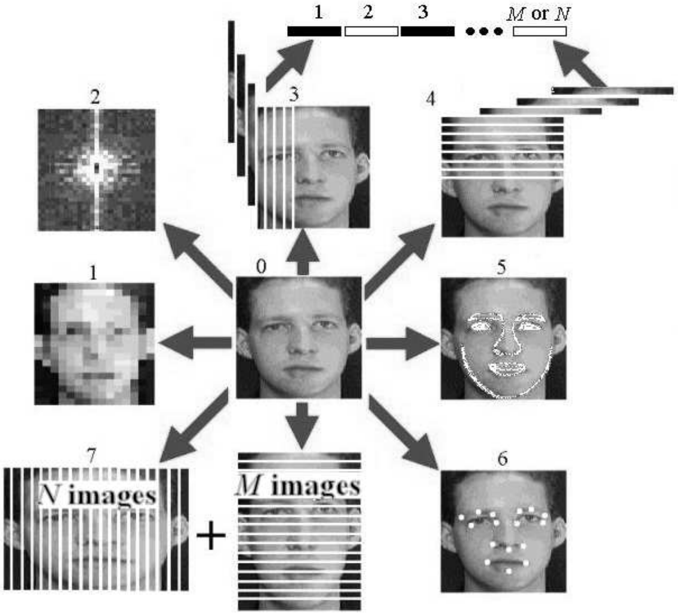


Fig. 1. Features acquirement.

Method	1	2	3	4	5	6	7
DIM	mn	200	MN	MN	≈ 200	≈ 100	$max\{M, N\}$

Tab. 1. Comparison of input features dimensionality.

The input dataset (face database – FDB) is described using the following parameters: K - class number, Q - the number of members of each class (the same for all classes), L - the number of elements used for training, $(Q-L)$ – the number of images used for testing $(Q-L\geq 1)$. Each member is an image stored in a matrix $O_{M\times N}$, where M,N are constant for all images. Input data are described using a matrix of $DIM\times 1$ features, where DIM is a number of features before reduction(s). In most cases, when the reduction is effective, $DIM\leq MN$. Reduction process can be carried out for the whole dataset or for each class separately. In the first case, we mean "Global PCA" (called GPCA). In the second case – "Local PCAs" (LPCA, [7]). Reduction based on LDA can be performed only for the whole dataset (for all classes). It is used as a single stage method or together with PCA/KLT [6, 12, 13, 14, 15, 19]. On the other hand, images to be compared have $DIM\leq MN$, and their dimensions after the reduction stage are equal to p and s, where p and s are the dimensions of vectors after PCA/KLT and LDA respectively. Possible scenarios are presented in Figure 2.

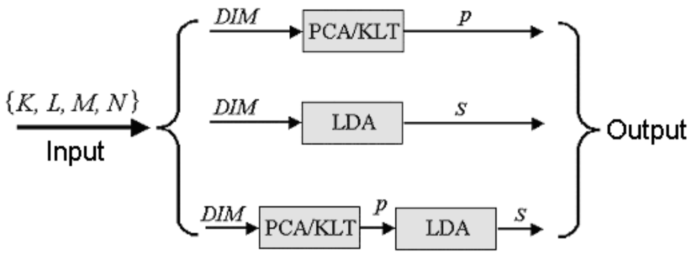


Fig. 2. Parameters of feature reduction system for PCA/KLT and LDA.

Proper work of a recognition system depends on the values of K,L,M,N,DIM, which should satisfy the following [6]:

$$KL>(DIM+K)>p>s,\ \text{for}\ DIM\leq MN\ \text{and}\ s\leq\quad K-1. \tag{1}$$

If the features to be reduced were created by concatenation of rows (or columns) of images from a database, then the following condition is true:

$$MN=DIM. \tag{2}$$

In our experiments we have used Olivetti Research Lab (ORL) face database [1], for which: $M=112$ and $N=92$. It gives $DIM=M\times N=10304$, but the whole database contains only 400 pictures (40 classes, 10 pictures per class), which is not enough. In the case of small values of K and L associated with large values of M and N $(KL<<MN)$, DIM should be changed to satisfy (1) using one of the following approaches (as shown in Figure 3) [9, 13, 14]:

1. Make M and N smaller, or

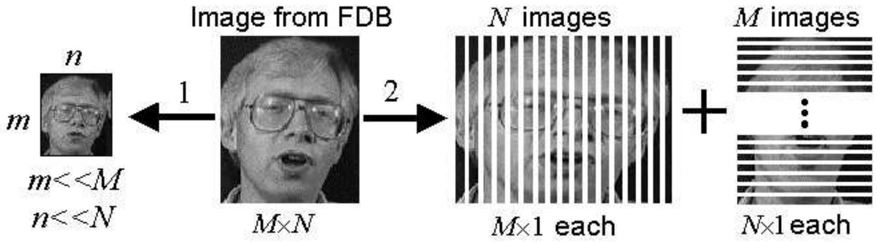


Fig. 3. Two ways of forming input data.

2. Change the structure of data by decomposition of input images into a set of row and column representations.

In the first approach we choose such values of m and n that $m << M$, $n << N$, and $DIM \equiv mn \leq KL$. In the second approach, each input image from the database is transformed into vectors by dividing it into N column-images (each of $M \times 1$ elements) or M row-images (each of $N \times 1$ elements). It formally increases the number of images to LN or LM, and the dimensionality of feature space becomes $DIM = \max\{M, N\} \leq KL(M + N)$.

In the process of synthesis of a recognition system the first step is to divide the dataset into two sub-sets: one for training (learning) and the other for testing (and tuning). The method image selection is based on the dispersion value. Images with high deviation (very different from the mean image) are used for learning, which provides a good representation of the data.

1. Therefore, we evaluate the mean image:

$$\bar{O}^{(k)}_{M\times N} = \frac{1}{Q}\sum_{q=1}^{Q} O^{(k,q)}_{M\times N},\ k = 1, 2, 3, \ldots, K \tag{3}$$

and deviation $\Delta(q)$ from $\bar{O}^{(k)}_{M\times N}$ for each image in the database:

$$\Delta(q) = \sqrt{(O^{(k,q)}_{M\times N} - \bar{O}^{(k)}_{M\times N})^2}, q = 1, 2, 3, \ldots, Q. \tag{4}$$

2. In the next step, we sort images in a decreasing order of $\Delta(q)$, so $\Delta(1) > \Delta(2) > \Delta(3) > \ldots$ and move them into a new database with memorized deviations. This database will be called "virtual database".
3. We select first L images of each class ($L < Q$) from the database for learning and the rest of them $(Q - L) \geq 1$ for testing.

3. Reduction

3.1. Preliminary Reduction Stage

Input data consists of many images, each image being represented by a matrix of many multi-byte elements. There are two basic methods of representing input data. The first one is to concatenate rows or columns of an image into a vector. The second one - to decompose the image into row and column representation [9], which will preserve important information about pixels' 2D positions.

The first reduction of input data is achieved by PCA/KLT [9, 14, 19]:

1. Based on images selected for the learning stage, we calculate the mean image according to:

$$\bar{O}_{M\times N} = \frac{1}{LK}\sum_{k=1}^{K}\sum_{q=1}^{L} O_{M\times N}^{(k,q)}. \tag{5}$$

2. According to a classical method, covariance matrices $R^{(1)}$ calculated for images of $M \times N$ size, have an order equal to $DIM = MN$. This approach can be simply described as:

$$\begin{array}{c} R^{(1)} = DD^T; \\ \Lambda^{(1)} = V^T R^{(1)} V, \end{array} \tag{6}$$

where:
D – feature matrix for input data, where each column (of size DIM) carries one image in the form of a vector. There are $K \times L$ column – images in the database;
$R^{(1)}$ – covariance matrix of order DIM;
$\Lambda^{(1)}, V$ – matrices (of order DIM) of eigenvalues and eigenvectors, respectively.

This method causes covariance matrices to be large and not easy to handle. It is also complicated to calculate eigenvalues for this kind of large matrices. The whole problem was presented in [9]. That is why we construct covariance matrices: R_M and C_N of M-th and N-th order, where the first one is calculated along the rows of each image and the second one - along the columns (this modification of PCA/KLT is called PCArc [9, 14]):

$$R_M = \frac{1}{LK}\sum_{k=1}^{K}\sum_{q=1}^{L}(O_{M\times N}^{(k,q)} - \bar{O}_{M\times N})(O_{M\times N}^{(k,q)} - \bar{O}_{M\times N})^T, \tag{7}$$

and

$$C_N = \frac{1}{LK}\sum_{k=1}^{K}\sum_{q=1}^{L}(O_{M\times N}^{(k,q)} - \bar{O}_{M\times N})^T(O_{M\times N}^{(k,q)} - \bar{O}_{M\times N}). \tag{8}$$

For each matrix (7) and (8) we can calculate eigenvalues and eigenvectors, for which the following is true:

$$\left.\begin{array}{l} \Lambda_M^{(R)} = [V_M^{(R)}]^T R_M V_M^{(R)} \\ \Lambda_N^{(C)} = [V_N^{(C)}]^T C_N V_N^{(C)} \end{array}\right\}, \tag{9}$$

where:
$\Lambda_M^{(R)}, \Lambda_N^{(C)}$ – diagonal matrices of eigenvalues,
$V_M^{(R)}, V_N^{(C)}$ – orthogonal matrices, whose columns contain eigenvectors;

Diagonalization of (9) is possible only when $V_M^{(R)}$, $V_N^{(C)}$ are orthogonal, which can be described as:

$$\left.\begin{array}{l} V_M^{(R)}[V_M^{(R)}]^T = I_M \\ V_N^{(C)}[V_N^{(C)}]^T = I_N \end{array}\right\}, \tag{10}$$

where: I_M, I_N – identity matrices of M-th and N-th order.

On the diagonals of $\Lambda_M^{(R)}$ and $\Lambda_N^{(C)}$ we select p largest elements and memorize their positions; according to the preliminary assumptions $p << DIM$ (which describes the order of feature matrix). From the matrix $[V_M^{(R)}]^T$ we select p rows and from the matrix $V_N^{(C)}$ – p columns corresponding to the p largest eigenvalues. Then we construct matrices $F_{p\times M}^{(R)}$ and $F_{N\times p}^{(C)}$ used later for reduction (this matrices are called "KLT matrices").

In Figure 4 the mean image for the sample database $\{K = 20, L = 7\}$ and the largest eigenvalues are presented. After the reduction, each image is represented by p^2 elements. Final reduced feature space is, compared to the input space, MN/p^2 times smaller, where M and N are dimensions of the input images. For each image from the part of the database used for learning we perform KLT according to the following formula:

$$Y_p^{(k,q)} = F_{p\times M}^{(R)}(O_{M\times N}^{(k,q)} - \bar{O}_{M\times N})F_{N\times p}^{(C)}, \left\{\begin{array}{l} \forall k = 1, 2, \ldots, K; \\ \forall q = 1, 2, \ldots, L; \end{array}\right. \tag{11}$$

where:

$$Y_p^{(k,q)} = \begin{bmatrix} y(1,1) & \ldots & y(1,p) \\ \vdots & \ddots & \vdots \\ y(p,1) & \ldots & y(p,p) \end{bmatrix}^{(k,q)}; \tag{12}$$

$y(i, j)$ – value of a feature,
p – size of reduced feature space, $p << DIM = \max\{M, N\}$.

The process of reduction is presented in Figure 5.

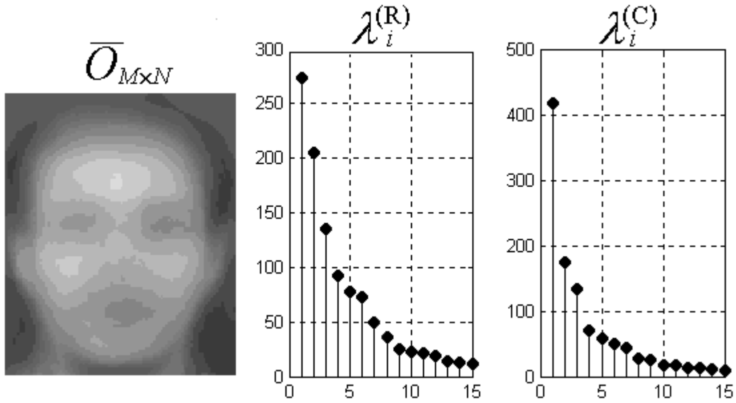


Fig. 4. Intermediate results of PCArc.

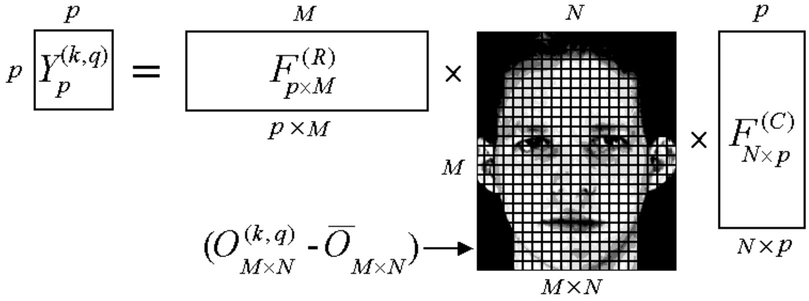
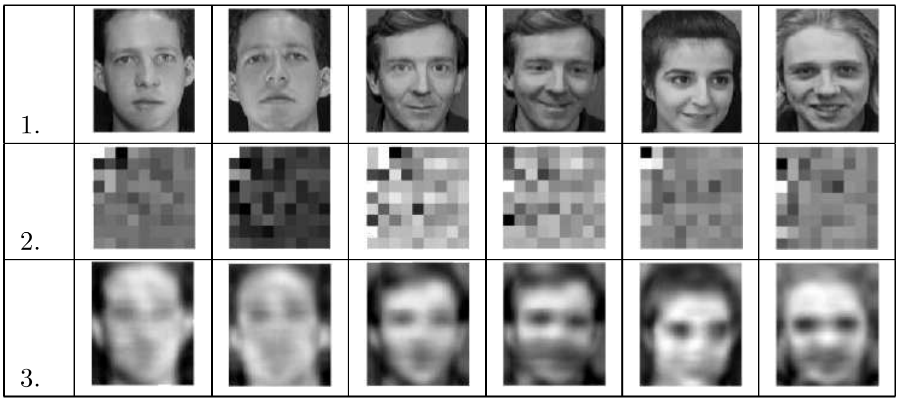


Fig. 5. Feature reduction scheme for each image.

Table 2 presents sample images from the database and their reduced representations together with reconstructed images, the upper row contains an input image, the middle row – its KLT spectrum (reduced features), and the lower row - a reconstructed image. The size of input data is 112×92 pixels, while reduced features occupy only a 9×9 matrix, which gives compression ratio of over $120 : 1$.

Projection of reduced feature set into a 3D space for a database described by $\{K = 20; L = 7; p = 9\}$ is shown in Figure 6. The first three components $(Y(1,1), Y(1,2), Y(2,1)$ are assigned to axes X, Y and Z respectively. Centres of classes are marked with vertical lines while the elements of each class are marked with dots connected with the apropriate centre by a solid thin line.



Tab. 2. Images (1), their two-dimensional KLT representations (2) and reconstucted images (3).

The mean value of all images in class k can be also calculated using a different approach:

$$\bar{O}^{(k)} = \frac{1}{LMN} \sum_{q=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N} O_{M\times N}^{(k,q)}, \text{ where } \bar{O}^{(k)} \text{ is a scalar.} \tag{13}$$

The mean value (13) can be used in all formulas above and it does not change the main principle of PCArc. In the approach described in the beginning, we selected p rows from $\left[V_M^{(R)}\right]^T$, and p columns from $V_N^{(C)}$, which have been used for construction of $F_{p\times M}^{(R)}$ and $F_{N\times p}^{(C)}$ (KLT matrices). We could have select also p_1 rows and p_2 columns, where $p_1 \neq p_2$. It will result in a rectangular matrix $Y_{p_1\times p_2}^{(k,q)}$ of $p_1 \times p_2$ elements. After such a reduction, each image will be represented by $p_1 p_2$ elements, which gives a reduction ratio of $MN/(p_1 p_2)$. Two above mentioned propositions related to the mean image calculation method and parameter (p_1, p_2 versus p) selection can be used to modify the PCArc algorithm. As it has been shown, PCArc is simple to interpret, implement and analyze. On the other hand, it does not require a large amount of calculations, because the order of a covariance matrix is not larger than $\max\{M, N\}$. PCA/KLT can successfully be used for feature dimensionality reduction, but it can not deal with clusterization tasks. It is also true for PCArc/KLT. Figure 6 shows that distances between classes are not maximized and distances inside each class are not minimized. It means that some images are placed far from centres of their classes and classes are not evenly spread in the whole space. After PCArc reduction, which can be described as:

$$O_{M\times N}^{(k,q)} \xrightarrow{\text{PCArc}} \left\{Y_p^{(k,q)}, F_{p\times M}^{(R)}, F_{N\times p}^{(C)}, \bar{O}\right\}, \left\{ \begin{array}{l} \forall k = 1, 2, \ldots, K; \\ \forall q = 1, 2, \ldots, L, \end{array} \right. \tag{14}$$

we have to store matrices of the KL transform (11) together with the mean image calculated according to (5) or the mean value calculated according to (13).

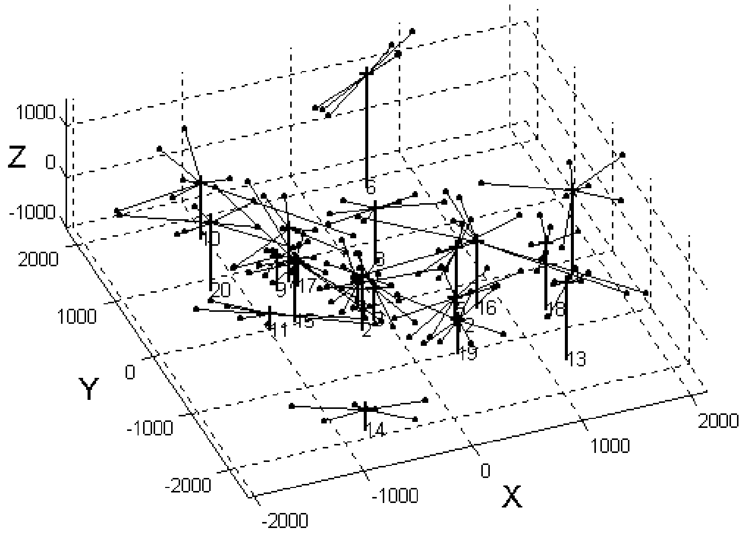


Fig. 6. Projection of features into 3D space after the first reduction.

For $DIM >> 100$, accuracy of calculating the eigenvalues and eigenvectors can not be guaranteed, and for $DIM >> 400$ numerical implementation is not easy. On the other hand, an image with dimensions $MN \leq 100$ is not suitable for recognition if there are many classes. This all makes traditional covariance matrix calculation unsuitable in this case. We will now present a different approach to covariance matrix, eigenvalues and eigenvectors calculation. If $KL << DIM$, then the whole process can be described as: [6, 12]:

$$\begin{gathered} R^{(2)} = D^T D; \\ \Lambda^{(2)} = W^T R^{(2)} W; \\ V^{(2)} = DW, \end{gathered} \tag{15}$$

where:
D – a matrix (of size $DIM \times KL$) of input images' features
$R^{(2)}$ – a covariance matrix of KL order ;
$\Lambda^{(2)}, W$ – matrices (of KL order) of eigenvalues and eigenvectors;
$V^{(2)}$ – a matrix (of size $DIM \times KL$) of eigenvectors.

Using (15) we can calculate the $\Lambda^{(2)}$ (matrix of eigenvalues) which, through additional multiplication, can give $V^{(2)}$ of eigenvectors, eqivalent to (6):

$$v^{(2)}_{(i,j)} \equiv v_{(i,j)}, \begin{cases} \forall i = 1, 2, \ldots, DIM \\ \forall j = 1, 2, \ldots, KL \end{cases}, \tag{16}$$

where $v^{(2)}_{(i,j)}$, $v_{(i,j)}$ – the element in i-th row and j-th column of $V^{(2)}$ and V respectively.

Dimensionality reduction is performed based on KLT for all input images:

$$Y_{p\times KL} = F_{p\times DIM} D_{DIM\times KL}, \tag{17}$$

where:

$$Y_{p\times KL} = [Y^{(1)} Y^{(2)} \ldots Y^{(KL)}], \tag{18}$$

$$Y^{(q)} = \begin{bmatrix} y(1,q) \\ y(2,q) \\ \vdots \\ y(p,q) \end{bmatrix}, \forall q = 1, 2, \ldots, KL; \tag{19}$$

where:
$Y^{(q)}$ - q-th image in reduced feature space;
$y(i, q)$ - i-th feature for q-th image, $i = 1, 2, ...p$;
p - size of the reduced feature space ($p << KL$);
$F_{p\times DIM}$ – a transformation matrix which is constructed from p rows of $V^{(2)}$
corresponding to the largest p eigenvalues of $\Lambda^{(2)}$.

The comparison of algorithms for computing matrices (6) and (15) reveals that the second approach is better because:

1. The covariance matrix has the order equal to KL, which decreases the computational complexity;
2. KL is much smaller than $DIM = MN$, which makes success in eigenvalues and eigenvectors calculation more probable.

3.2. Second Reduction Stage

To improve clusterization effects (resulting in the "compactness" of the dataset) the second stage of reduction is introduced. The idea is presented in [6]. We perform linear discriminant analysis on PCArc-processed data set. The aim of this stage is to make elements of each class close to each other within the class and to spread classes evenly in the feature space.

Matrices (11) should be stored in vectors of size $p^2 \times 1$, which are obtained by concatenation of columns or rows of $Y_p^{(k,q)}$. These vectors will be later described as $X^{(k,q)}$ for each $k = 1, 2, ..., K$ and $q = 1, 2, ..., L$. We calculate the mean value for all elements of the feature set and mean value $\bar{X}^{(k)}$ for each class $k = 1, 2, ..., K$ separately:

$$\bar{X} = \frac{1}{KL} \sum_{k=1}^{K} \sum_{q=1}^{L} X^{(k,q)}, \tag{20}$$

$$\bar{X}^{(k)} = \frac{1}{L} \sum_{q=1}^{L} X^{(k,q)}, K = 1, 2, \ldots, K. \tag{21}$$

The way of calculating covariance matrices is similar to the one used in the PCA/KLT. We have two covariance matrices: W_{p^2} – the within-class covariancematrix and B_{p^2} – between-class covariance matrix:

$$W_{p^2} = \sum_{k=1}^{K} \sum_{q=1}^{L} (X^{(k,q)} - \bar{X}^{(k)})(X^{(k,q)} - \bar{X}^{(k)})^T, \tag{22}$$

$$B_{p^2} = \sum_{k=1}^{K} (\bar{X}^{(k)} - \bar{X})(\bar{X}^{(k)} - \bar{X})^T. \tag{23}$$

Based on (22) and (23) we calculate a common covariance matrix:

$$H_{p^2} = W_{p^2}^{-1} B_{p^2}, \tag{24}$$

for which we calculate eigenvalues and eigenvectors:

$$\Omega_{p^2} = \left[U_{p^2}\right]^T H_{p^2} U_{p^2}, \tag{25}$$

where:
Ω_{p^2} – the diagonal matrix of of eigenvalues;
U_{p^2} – the orthogonal matrix, whose rows contain the eigenvectors.

From the diagonal of Ω_{p^2} we select s largest elements and memorize their positions $(s \leq K - 1, s \neq 0)$. From $\left[U_{p^2}\right]^T$ we select s rows corresponding to s largest eigenvalues and we build a new matrix $A_{s \times p^2}$ from them, which will be used as a reduction matrix. The feature space $X^{(k,q)}$ for $k = 1, 2, \ldots, K$ and $q = 1, 2, \ldots, L$, is reduced according to the following formula:

$$\hat{X}^{(k,q)} = A_{s \times p^2} X^{(k,q)}, \tag{26}$$

where each vector $\hat{X}^{(k,q)}$ is of size $s \times 1$.

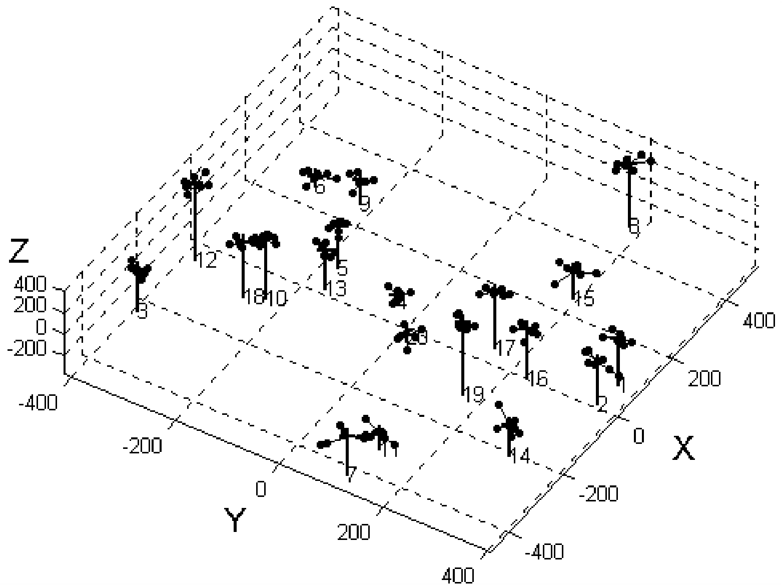


Fig. 7. Feature space after second reduction in 3D representation.

Projection of reduced features (for: $K = 20; L = 7; p = 9$ and $s = 19$) into the 3d space is presented in Figure 7. The axes X, Y and Z are assigned to the first three elements of the vector $\hat{X}^{(k,q)}$. LDA gives further reduction and, as it can be seen, improves clusterization effects. The centres of classes are evenly spread in the whole space. On the other hand, elements of each class are close to each other and close to the centre of their own class. This situation enables us to solve recognition tasks using rather simple methods, for example minimum distance metric.

After PCArc+LDA reduction we have:

$$O^{(k,q)}_{M\times N} \xrightarrow{\text{PCArc+LDA}} \left\{\hat{X}^{(k,q)}, A_{s\times p^2}, F^{(R)}_{p\times M}, F^{(C)}_{N\times p}, \bar{O}\right\}, \left\{\begin{array}{l}\forall k=1,2,\ldots,K;\\ \forall q=1,2,\ldots,L.\end{array}\right. \qquad (27)$$

For performing the reduction we have to store 3 matrices of KLT (10) and LDA (26) and the mean image calculated according to (5) or the mean value (13). It should be remembered that the vector (26) describing the whole image (face) consists of s elements only (its size is $s \times 1$). Reduced feature space is MN/s times smaller than the input one.

4. Recognition of Test Images

The process of recognition of test images can be presented on an example of the face recognition task. Different images of a few faces are stored in a database. Each image can be taken as a template. An image to be recognized is presented to the recognition system. Recognition is preformed by calculating some similarity coefficient between this image and each template. Recognized image is then assigned to a class, which has highest similarity coefficient but not lower than some threshold. Let us assume, that sample recognition system is developed for parameters $\{M, N, K, Q, L, p, s\}$. After the synthesis of the system, we have: vectors $\hat{X}^{(k,q)}$ of reduced feature space; reduction matrices $F^{(R)}_{p\times M}, F^{(C)}_{N\times p}, A_{s\times p^2}$ and the mean image or the mean value $\bar{O}$. The process of recognition of test images $O^{(k,q)}_{M\times N}$ for $q = L+1, L+2, \ldots, Q$ is performed in 3 stages:

1. Mean value (or mean image) removal from each image;
2. Features reduction using $F^{(R)}_{p\times M}, F^{(C)}_{N\times p}, A_{s\times p^2}$, which gives $\tilde{X}$;
3. Distance calculation between $\tilde{X}$ and mean vectors $\bar{X}^{(k)}$ for each class, and remebering the number of the class with the closest distance (which is equal to the highest similarity factor).

The first two stages can be described by the following equation:

$$\tilde{X}^{(k,q)} = A_{s\times p^2}\left(F^{(R)}_{p\times M}(O^{(k,q)}_{M\times N} - \bar{O}_{M\times N})F^{(C)}_{N\times p}\right), \tag{28}$$

where:

$\tilde{X}^{(k,q)}$ – a vector of size $s \times 1$;
$k = 1, 2, \ldots, K$;
$q = L+1, L+2, \ldots, Q$.

The distance between vectors is calculated using one of known metrics, for example, the Euclidean distance:

$$d_k = \sqrt{\left(\tilde{x}_1 - \bar{x}^{(k)}_1\right)^2 + \left(\tilde{x}_2 - \bar{x}^{(k)}_2\right)^2 + \ldots + \left(\tilde{x}_s - \bar{x}^{(k)}_s\right)^2}, \tag{29}$$

where:
$\tilde{x}_i, \bar{x}^{(k)}_i$ – elements of vector (28) and the center of k-th class;
$i = 1, 2, \ldots, s$;
$k = 1, 2, \ldots, K$.

We have chosen the Euclidean distance, as it is simple, easy to implement and in our case gave good enough results. The distance calculation from the center of a class can be followed by distance calculation within the chosen class to find the most similar image. But it should be remembered that it is not the only metric that can be used.

The PCArc/KLT/LDA method, which was presented above, includes approaches from [6, 9, 12] and from our point of view is optimal for face recognition. In addition, we present some results of experiments on recognition of human faces in the ORL database for systems with different parameters $\{M, N, K, Q, L, p, s\}$, and test images distorted by different factors: scaling, noise etc. For the first time, these results were discussed in [13, 19].

4.1. Change in The Class Number and The Member Number

The aim of the experiment carried out was to evaluate the influence the parameter K on the synthesis and robustness of the recognition system. We used $4, 20, 40$ and 49 as values of K. It should be pointed out that small number of classes (up to 10) is suitable only for "Visitor Identification" and "Access Control" tasks. Larger number of classes is suitable for retrieving images from face databases.

We performed some experiments on face recognition systems built using the presented algorithms with different values of parameters. The database contained images from the ORL face database in the form of gray-scale pictures of size 112×92 pixels.

For a simple recognition system (used in "Visitor Identification" tasks), where $K = 4$, it is necessary to have at least $L = 6$ pictures per person. For conditions presented above we achieved very high recognition rate (more than 95%).

In the case, when the number of classes K is higher, the number of images L should be increased as well. The value p should also be increased. That is why, all possible values of p should be checked in order to find the minimal value of p, for which the recognition rate is still satisfying. We chose the following parameters: $\{K = 20; L = 7; p = 9; s = 19\}$. For 60 test images (3 images per class) and a system described by $\{K = 20; L = 7; p = 9$ and $s = 19\}$ we achieved 100% accuracy. For a system with $\{K = 40; L = 7; p = 7, 8, 9, 10$ and $s = 39\}$ the recognition rate was also 100% [19].

All the experiments presented above were carried out on the ORL face database. It is possible, that the high recognition accuracy is strongly associated with the high quality of collected images. But it is also clear that the way the data is divided into test and training sets has some influence on that as well. It should be remembered that test images have lower deviations from their own class centre than ones used for learning.

4.2. Recognition of Noisy Images

Performing the recognition on noisy images (which were not filtered) can give information about the noise-robustness of developed algorithm. In our experiments we tested the recognition rate of a sample system described by $\{K = 20; L = 7; p = 9$ and $s = 19\}$ with test images that were distorted by white noise (see images presented in Figure 8). First 7 images of each class were used for synthesis (learning). The rest (3 test images) were influenced by white noise of varying amplitude $x \geq 100$ and uniform distribution

$\delta = 1$ (from the intensity range 0-255).

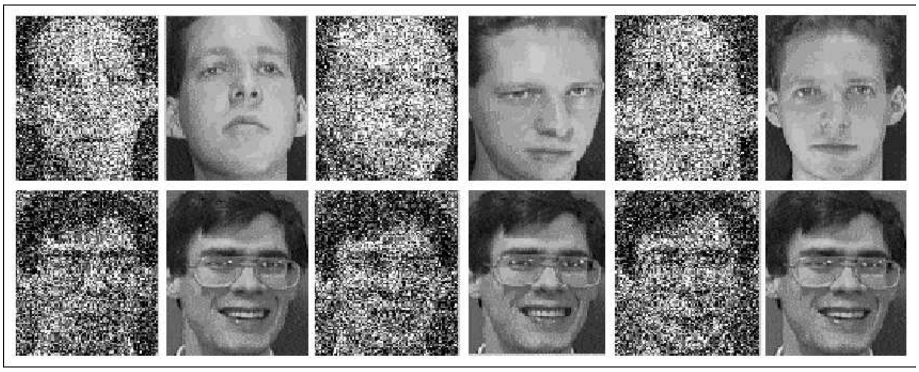


Fig. 8. Recognition of noise-influenced images.

Tables 3 and 4 below present the recognition rate for two systems described by: $\{K = 20; Q = 10; L = 7; p = 7$ and $s = 19\}$ and $\{K = 40; Q = 10; L = 7; p = 12; s = 39\}$ respectively. Is is possible to increase the recognition rate by majority voting (e.g. choosing 2 images out of 3 recognized) which is presented in the last row of tab. 3 and tab. 4. The comparison of the "two-out-of-three" principle (which is identical to "k Nearest Neighbours" method) with unvoted approach is shown in Figure 9 as well.

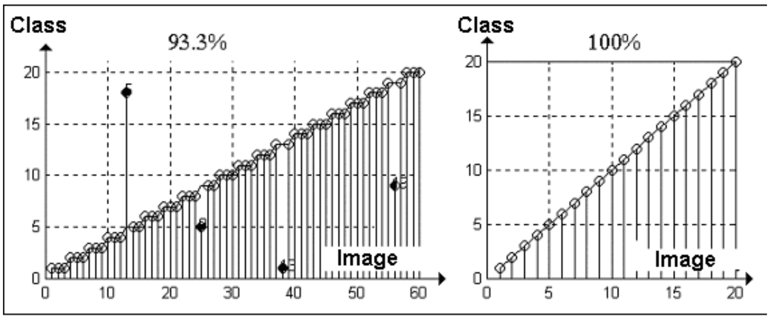


Fig. 9. Two ways of recognition results presentation.

Analogous experiments were carried out for different values of K. They showed that the successful recognition is guaranteed for the noise amplitude up to 100. It should be stressed that such extreme noise levels are not common in practice. On the other hand, we can always introduce some filtering stage to improve the image quality.

noise amplitude	60	70	80	90	100	110	120	130
unvoted	100	100	100	100	100	100	100	98
two-out-of-three	100	100	100	100	100	100	100	100

Tab. 3. Results of recognition of 60 noisy images ($K = 20$).

noise amplitude	60	70	80	90	100	110	120	130
unvoted	100	100	99	98	96	95	95	88
two-out-of-three	100	100	100	100	100	100	97	95

Tab. 4. Results of recognition of 120 noisy images($K = 40$).

4.3. Recognition of Blurred Images

In practice there exists a problem of recognizing blurred or de-focused images. It happens frequently during image acquisition when an optical sensor does not properly focus on the object or the analyzed image is a small fragment of a large image. By down-scaling the images by factor $a \in \{1/2, 1/3, 1/4, 1/5, 1/6\}$, and then re-scaling them back to the original size (by factor 2,3,4,5 and 6 respectively) using bilinear or bicubic interpolation we get an effect similar to changing the focal length. In addition, a simple nearest-pixel scaling was also tested. It is based on copying the nearest pixel value into the place, where there is a gap after size change. It leads to repetition of certain pixels M/m times along the rows and N/n times along the columns. Comparison of these methods is presented in Figure 10. A recognition system was synthesized for the following parameters: $\{K = 40, Q = 10, L = 7, p = 12, s = 39\}$. Features reduction was carried out on intensity images (in gray scale) of size $M \times N$. Test results for interpolated images are in Table 5 and Table 6. Again, when we want to improve recognition accuracy, we use ”two-out-of-three” principle. The results of this approach are presented in the lower row of each table. Each interpolation type introduces certain errors, which can be seen in Figure 10. The presented image (112 × 92 pixels) was down-scaled 6 times and then up-scaled using 3 types of interpolation. Coarsely quantized images are obtained by nearest-pixel interpolation. It involves scaling by 1/2, 1/3, 1/4, 1/5 and 1/6 times and then restoring initial dimensions. The results of recognition of quantized images (see Figure 11) are presented in Table 7.

scale	1	1/2	1/3	1/4	1/5	1/6
unvoted	100	100	100	98	98	92
two-out-of-three	100	100	100	100	100	95

Tab. 5. Summary of the recognition of 120 interpolated (bi-cubic) images.

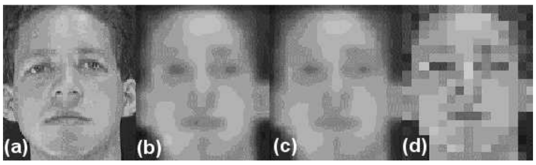


Fig. 10. Comparison of interpolation methods used for restoring the required image size: (a) original image,(b) bi-linear interpolation, (c) bi-cubic interpolation, (d) the nearest-pixel method.

scale	1	1/2	1/3	1/4	1/5	1/6
unvoted	100	100	100	100	97	92
two-out-of-three	100	100	100	100	97	95

Tab. 6. Summary of the recognition of 120 interpolated (bi-linear) images.

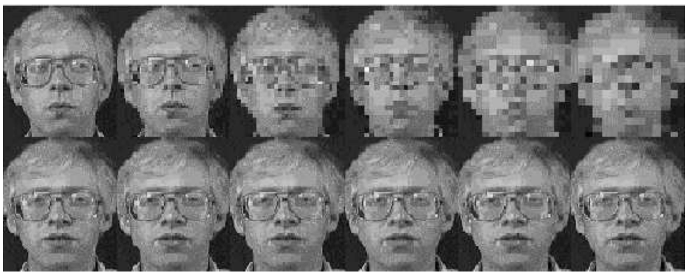


Fig. 11. Results of the recognition of coarsely quantized images. Upper row – images to be recognized, lower row – images retrieved from database by means of our algorithm.

scale	1	1/2	1/3	1/4	1/5	1/6
unvoted	100	100	100	99	98	97
two-out-of-three	100	100	100	100	97	97

Tab. 7. Summary of the recognition of coarsely interpolated images.

The experiments have shown that interpolation involved in re-scaling of small images to the reference size influences the recognition rate to some extent. We can assume that successful recognition can be achieved with scaling not lower than $a = 1/4$. Recognized images should also be bigger than about 28 × 23 pixels [19].

4.4. Recognition of Distorted Images

In real conditions we deal with distorted images. It is caused by lossy compression algorithms, optical acquisition, scanning, sampling, etc. The input image can be either smaller, than it is required, and/or can be noisy. However, all the factors can not easily be foreseen. For our experiments we have assumed, that the distortions can be grouped (in 6 groups) to simplify testing. Grouping method is presented in Table 8. Signs "+" and "-" inform about the presence or absence of noise and signs "↑" and "↓" stand for up-scaling and down-scaling of an image to meet the database requirements. Figure 12 shows 2 groups of distorted images (belonging to the same class). The distortion parameters (scale factor and noise level) for the first group (three images on the left) are $a = 1/4, x = 20$ and for the second group – $a = 1/4, x = 50$ respectively.

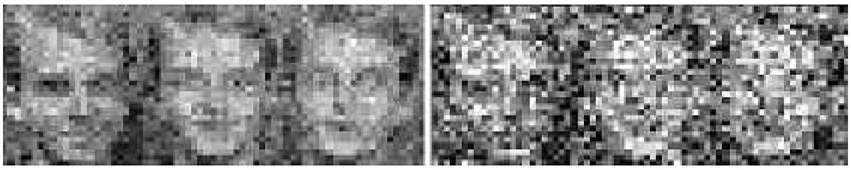


Fig. 12. Sample images belonging to the same class under complex distortions (scaling and noise).

	image size	noise before	scaling	noise after
1	smaller	-	↑	+
2	smaller	+	↑	-
3	smaller	+	↑	+
4	larger	-	↓	+
5	larger	+	↓	-
6	larger	+	↓	+

Tab. 8. Possible changes in test images.

The experiments show that correct recognition of distorted images is possible for $a = 1/3$ and $x = 40$, or for $a = 1/4$ and $x = 20$. The value of x should not be larger than 20. Similar values of a and x were obtained for test images from the whole ORL database, which consists of 40 different classes [19].

To summarize the experiments we can say that analyzed methods (PCArc/KLT + LDA) based on [6, 9, 12] are suitable for recognition of noised and scaled images. The recognition rate of 100% can be reached for different values of x and a. That is why, the algorithms are suitable for practical implementations.

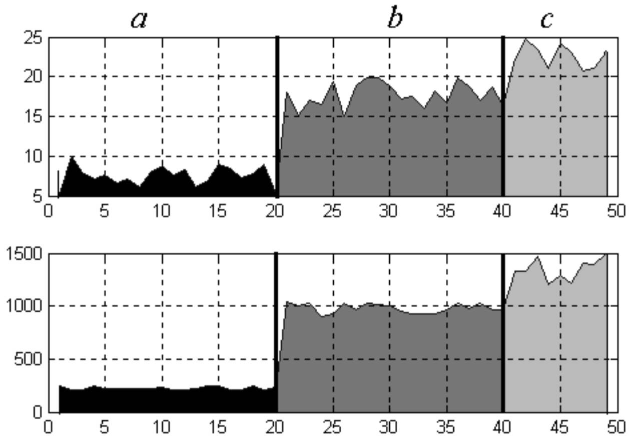


Fig. 13. Results of distance calculation for images from groups (a), (b) and (c).

4.5. Recognition of Images Not Present In Database

In our experiments we have tried to simulate real situations, when the image to be recognized can not be found in the database. It means that an image we want to recognize belongs to a person who has not been included in our system. It is clear that this case a classification system will find some "nearest" class, but it will not be a successful recognition. In literature this kind of situation is called "False Acceptance" and a percentage of false recognized images is known as "False Acceptance Rate". Therefore we have tried to answer the following two questions:

1. How can the results of recognition help in distinguishing between an in-database-image and out-of-database-image?
2. What is the criterion of similarity and dissimilarity in this approach?

To solve the above problems, we have performed the following experiment. Our recognition system $\{K = 20, Q = 10, L = 7, p = 7, s = 19\}$ was shown 60 test images belonging to classes 1-20 from ORL database, 87 images from classes 21-40 from ORL face database and 27 other images. In this experiment we used the similarity coefficient calculated as a distance (L_0 and L_2) from the centre of each class in the reduced feature space. For L_0 we set the threshold t to 30, that is 30% of the range of each feature. Then, because of the fact that the system was shown 3 test images, the final result is taken as the mean value of 3 single results. As a result, we have 49 values of distances for each metric (because (60+87)/3=49). These results are shown in Figure 13.

The upper diagram in Figure 13 shows the distance L_0 and the lower one – L_2. Letters "a", "b", and "c" mark different sets of images:

a) First 20 classes of the ORL database (they were used for building the database);

b) Classes from 21 to 40 from the ORL database;

c) Images from another database.

The results show that:

1) smallest distances were obtained for test images belonging to their own classes (”a” in diagram);

2) distances we used are suitable for distinguishing images belonging to their own database from images belonging to another one;

3) distance evaluation (using L_2) was successful for images ”b” and ”c” (collected from different sources).

Our analysis proves that L_0 metric does not show differences between images from different sources. The main reason for that are the parameters of the system: $\{K = 20, Q = 10, L = 7, p = 7, s = 19\}$ where all 19 components were used for distance calculation. Even in this case, the L_0 metric can distinguish images from their own database from images comming from a different database. Images in group ”a” have much smaller distances in comparison to images ”b” and ”c”.

To summarize: for verification purposes of distinguishing between ”own” and ”different” database, both metrics can be used, but L_2 gives much better results. To complete the experiment, we now show some results of recognition of images not present in the database. In Figure 14 there are 3 pairs of images shown. Images on the left are presented to a recognition system. Right-hand images are retrieved from the database. The class number is shown above the image (related to the ORL database).

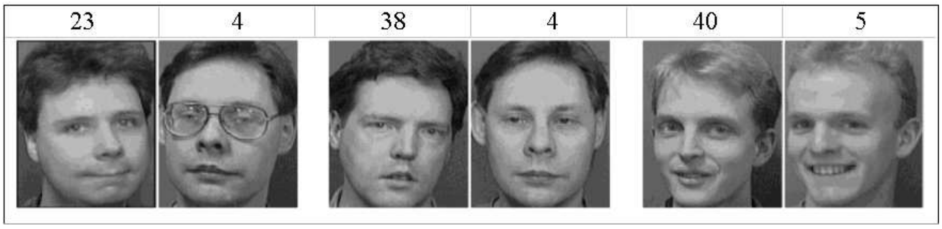


Fig. 14. Results of recognition of images not present in database.

Visual inspection of the images shown confirms that the implemented algorithm retrieves the ”closest” image from a database. Retrieved images have similar orientation of the head, coarse details of the face (eyebrows, eyes, glasses, beard, hair-style, hair color etc.), and a placement of the face in the image.

To compensate for such effects the threshold of positive recognition should be increased, so images, which are ”close” but do not belong to the selected class, are rejected.

4.6. Comparison with Classical Approach

To compare the presented method of building a database and recognizing sample images we constructed 2 face recognition systems. The first one was based on a classical approach (which implies scalling of input images to appropirate size and features reduction by means of basic PCA/KLT), the second one was based on the methods discussed. Sample results for systems involving the reduction to $p = 25$ for PCA/KLT and $p = 5 \times 5$ for PCArc are presented in Table 9. As it can be seen from this table (columns "Recog. Results (PCA)" and "Recog. Results (PCArc)"), the system which utilizes PCArc gave better results for the same number of features after reduction. This shows the advantage of PCArc method over classical data processing using PCA/KLT.

K	Q	L	$Q - L$	Training/ Testing Images	Size after Scalling	Recog. Results (PCA)	Recog. Results (PCArc)
40	10	7	3	280/120	16 x 15	99%	100%
40	10	6	4	240/160	15 x 13	96%	98%
40	10	5	5	200/200	14 x 12	93%	95%
40	10	4	6	160/240	13 x 11	91%	92%

Tab. 9. Comparision of two types of PCA/KLT-based recognition systems

5. Summary

The above examples show that calculating eigenvectors along rows and columns of an image and two-stage reduction (PCArc/KLT + LDA) of features dimensionality is suitable for different applications involving human face recognition. In the presented algorithm we use in parallel row and column representations of image to preserve information about the 2D stucture of the image. The cascade of PCArc and LDA leads to high compression levels and optimal representation in the feature space, which gives an efficient method of image recognition and browsing large graphical databases. It does not demend high computational power or large memory, which have been the well-known disadvantages of classical PCA.

We should also remember that nowadays all developed methods offer "almost" 100 percent of success in recognition. Most of them reach 90–95 percent. The method presented in this paper makes a small but important step towards the "real" 100 percent of success.

References

1994

[1] AT&T Laboratories Cambridge. The ORL Database of Faces. URL: www.uk.research.att.com/facedatabase.html

1995

[2] Swets D. and Weng J. : Efficient image retrieval using a network with complex neurons, in Proceedings, International Conference on Neural Networks, Perth, Western Australia, November.

[3] Swets D. L., Punch B. and Weng J. : Genetic algorithms for object recognition in a complex scene. ICIP, 2595–2598. URL: citeseer.nj.nec.com/swets95genetic.html

1996

[4] Belhumeur P.N., Hespanha J. and Kriegman D. J. : Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. ECCV, 45–58.

[5] Moses Y., Ullman S. and Edelman S. : Generalization to novel images in upright and inverted faces. URL: citeseer.nj.nec.com/article/moses94generalization.html

[6] Swets D. L. and Weng J. : Using Discriminant Eigenfeatures for Image Retrieval : IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, 831–836

1998

[7] Multimedia - Algorytmy i standardy kompresji [in Polish], Akademicka Oficyna Wydawnicza PLJ, Warszawa

[8] Nefian A. and Hayes M. : Hidden markov models for face recognition, in ICASSP98, 2721–2724. URL: citeseer.nj.nec.com/236258.html

[9] Tsapatsoulis N. , Alexopoulos V. and Kollias S. : A Vector Based Approximation of KLT and Its Application to Face Recognition. Proceedings of The IX European Signal Processing Conference EUSIPCO-98. Rhodos Palace, Island of Rhodes, Greece, September 8–11, 1581–1584

1999

[10] Liu C. and Wechsler H. : Comparative assessment of Independent Component Analysis (ICA) for face recognition. In Proc. the 2nd International Conference on Audioand Video-based Biometric Person Authentication, Washington D. C., March 22–24. URL: citeseer.nj.nec.com/liu99comparative.html

[11] Nefian A. and Hayes M. : Face recognition using an embedded HMM, in Proceedings of the IEEE Conference on Audio and Video-based Biometric Person Authentication, March, 19–24. URL: citeseer.nj.nec.com/nefian99face.html

[12] Swets D. L. and Weng J. : Hierarchical Discriminant Analysis for Image Retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 21, No.5, 386–401. URL: citeseer.nj.nec.com/swets99hierarchical.html

2000

[13] Kuchariew G. and Forczmanski P. : Kompresja i uporzadkowanie danych obrazowych dla zadan rozpoznawania [in Polish]. Materialy V Sesji Naukowej Informatyki, January, 15–21.

2001

[14] Kuchariew G. and Forczmanski P. : Hierarchical Method of Recuction of Features Dimensionality for Image Recognition and Graphical Data Retrieval. Proceedings of Sixth International Conference Pattern Recognition and Information Processing (PRIP), Minsk, Republic of Belarus, May, 19–34.

[15] Kuchariew G. and Tujaka A. : Pattern Recognition Methods for Visitor Identification and Access Control. Proceedings of Sixth International Conference Pattern Recognition and Information Processing (PRIP), Minsk, Republic of Belarus, May, 19-34.

[16] Lee T., Lewicki M. S., Sejnowski T. J. : Mixture Models for Unsupervised Classification of Non-Gaussian Classes and Automatic Context Switching in Blind Signal Separation. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 22, No. 10, 1078–1089. ULR: citeseer.nj.nec.com/lee00ica.html

[17] Martinez A. M.and Kak A. C. : PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 23, No. 2, 228–233. URL: citeseer.nj.nec.com/martinez01pca.html

2002

[18] Havran C., Hupet L., Czyz J., Lee J., Vandendorpe L. and Verleysen M. : Independent Component Analysis For Face Authentication. KES'2002 Processdings - Knowledge Based Intelligent Information and Engineering Systems, Crema, Italy, September. URL: citeseer.nj.nec.com/havran02independent.html

2003

[19] Kukharev G. and Kuzminski A. : Techniki biometryczne. Czesc I – Metody rozpoznawania twarzy [in Polish]. Wydzial Informatyki. Politechnika Szczecinska. Szczecin.