Appeared in journal: Neural Network World, Vol.6, No.3, 1996, pp. 375–382
Published by: IDG Co., Prague, Czech Republic

PRINCIPAL SUBSPACE ANALYSIS FOR INCOMPLETE IMAGE DATA IN ONE LEARNING EPOCH

Władysław SKARBEK[*], *Andrzej CICHOCKI*[†], *Włodzimierz KASPRZAK* [‡]

Abstract: In this paper we propose improved, high speed convergence algorithms for principal subspace analysis (PSA) and related principal component analysis (PCA). We have confirmed by computer simulations that applied recursive least squares (RLS) technique together with deflation preprocessing, dramatically improves the performance and reduces the training time to only one epoch for natural images. Furthermore, we have found that the training set can be reduced even to 10% of the total number of pixels, for high resolution images, without substantial loss of accuracy.

1. Introduction

Adaptive feature extraction is useful in many information processing systems. One of such tools for feature extraction is PCA and PSA [1, 2, 3, 4, 5]. Unfortunately, most of the known adaptive algorithms for PSA/PCA are relatively slow [2]. In this paper we propose an extension of learning algorithms which improves the convergence speed, especially for image compression.

2. Problem formulation

The *Principal Component Analysis* (PCA) problem can be formulated as follows: For the given dimensionality N and a zero-mean vector random variable $\boldsymbol{X}$, find a unit vector $\boldsymbol{w}$ such that the projection of $\boldsymbol{X}$ onto the line $\{t \cdot \boldsymbol{w} | t \in R\}$ is the scalar random variable of maximum variance, i.e.:

$$E_x \left[\left(\boldsymbol{w}^T \boldsymbol{x} \right)^2 \right] \quad \text{is maximum over the unit sphere } S_N \subset R^N. \tag{1}$$

[*] Władysław SKARBEK
Polish Academy of Sciences, Institute of Computer Science, Poland
[†] Andrzej CICHOCKI
Department of Electrical Engineering, Warsaw University of Technology, Poland
[‡] Włodzimierz KASPRZAK
FRP RIKEN, ABS Laboratory, 2–1 Hirosawa, Wako–shi, Saitama 351–01, Japan

The solution of the above problem is the normalized eigenvector (the *principal vector*) corresponding to the largest eigenvalue of the covariance matrix $E[\boldsymbol{x}\boldsymbol{x}^T]$.

The well-known Oja's learning rule [2] allows to find the principal vector:

$$\Delta \boldsymbol{w}_k = \eta_k y_k (\boldsymbol{x}_k - y_k \boldsymbol{w}_k), \quad \text{where} \quad y_k = \boldsymbol{w}_k^T \boldsymbol{x}_k \tag{2}$$

The convergence speed is controlled by the learning rate $\eta_k > 0$ which generally can be changed at every learning step. Usually many epochs are required to reach the solution (see e.g. Cichocki,Unbehanen [3]).

In order to improve the convergence speed we can apply recursive least squares (RLS) technique ([5, 6]) as follows:

$$\begin{aligned} \eta_0' &= E[\|\boldsymbol{x}\|^2] \\ y_k &= \boldsymbol{w}_k^T \boldsymbol{x}_k \\ \eta_k' &= \eta_{k-1}' + y_k^2 \\ \Delta \boldsymbol{w}_k &= (y_k/\eta_k')(\boldsymbol{x}_k - y_k \boldsymbol{w}_k) \\ \boldsymbol{w}_{k+1} &= \boldsymbol{w}_k + \Delta \boldsymbol{w}_k \end{aligned} \tag{3}$$

where $\eta' = 1/\eta$.

As our experiments show, for images of natural scenes this scheme finds the solution in one learning epoch with the error less than 0.1% when compared with results obtained by numerical methods. We refer further to the scheme (3) by RLS-Oja learning rule.

The algorithm (3) was used by the authors in [6] to determine K consecutive principal components by a deflation process: first time for the variable $\boldsymbol{X}_1 \doteq \boldsymbol{X}$ obtaining the principal vector $\boldsymbol{w}_1$, second time for the variable $\boldsymbol{X}_2 \doteq \boldsymbol{X}_1 - (\boldsymbol{w}_1^T \boldsymbol{X}_1)\boldsymbol{w}_1$ obtaining the vector $\boldsymbol{w}_2$, ..., and the K-th time for the variable $\boldsymbol{X}_K \doteq \boldsymbol{X}_{K-1} - (\boldsymbol{w}_{K-1}^T \boldsymbol{X}_{K-1})\boldsymbol{w}_{K-1}$.

Closely related to the PCA problem is the *Principal Subspace Analysis* (PSA) problem:
For given dimensionalities $1 \leq K < N$ and a zero-mean random vector variable $\boldsymbol{X}$, find a K-dimensional subspace of R^N, i.e. an orthonormal base $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$, such that the projection of $\boldsymbol{X}$ onto the subspace $span\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K\}$, is a K-dimensional random vector variable of maximum variance, i.e.:

$$E_x \left[\sum_{i=1}^{K} \left(\boldsymbol{w}_i^T \boldsymbol{x} \right)^2 \right] \quad \text{is maximum over all } K \text{ element orthonormal bases} \tag{4}$$

Introducing the $K \times N$ matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]^T$, we have an equivalent matrix formulation of the PSA problem:

$$E_x \left[\left\| \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{x} \right\|^2 \right] \quad \text{is maximum over all feasible } \boldsymbol{W} \tag{5}$$

From the Pythagorean theorem

$$\|\boldsymbol{x}\|^2 = \|\boldsymbol{x} - \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{x}\|^2 + \|\boldsymbol{W}^T \boldsymbol{W} \boldsymbol{x}\|^2 \ .$$

we obtain an equivalent, more convenient, minimization problem:

$$E_x \left[\left\| \boldsymbol{x} - \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{x} \right\|^2 \right] \quad \text{is minimum over all feasible } \boldsymbol{W} \tag{6}$$

For real data where space symmetries are less probable, usually there is a unique subspace, called the KLT (Karhunen-Loeve Transform) subspace which is the solution of the PSA problem. However, always there is an infinity of orthonormal bases for the KLT subspace.

The PSA problem has found many applications among which the image compression is one of the most important, as shown for instance in Skarbek [7].

A neural network solution for the PSA problem is based on Oja's subspace rule ([2, 8, 9]):

$$\Delta \boldsymbol{W}_k = \eta_k \boldsymbol{y}_k \left(\boldsymbol{x}_k - \boldsymbol{W}_k^T \boldsymbol{y}_k \right)^T, \quad \text{where} \quad \boldsymbol{y}_k = \boldsymbol{W}_k \boldsymbol{x}_k \tag{7}$$

Recently Karhunen [9] has shown that if we set $\eta_k \leq 2/\|\boldsymbol{x}_k\|^2$ and the initial weight matrix is suitably chosen, then the sequence of weight matrices produced by the rule (refojaPSA) is bounded. However, it implies only that a matrix subsequence is convergent.

The goal of this paper is to check whether RLS scheme for updating the learning rate η_k, developed for PCA, can be extended to the multidimensional case of PSA and to related PCA algorithms ([4, 6]).

3. Improved PSA algorithm

Let us consider again the RLS-Oja learning rule for PCA given by the equations (3). When transforming from PCA to PSA firstly we have to replace the scalar variable $y = \boldsymbol{w}^T \boldsymbol{x}$ by the vector variable $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$. In the next step the expression y^2 is replaced by the outer product $\boldsymbol{y}\boldsymbol{y}^T$. It means the scalar η' is generalized to a symmetric positive definite matrix parameter $\boldsymbol{P}'$. The matrix inverse $(\boldsymbol{P}')^{-1}$ will correspond to inverse of η'.

In an intermediate step our improved PSA learning rule (which is a generalization of RLS-Oja learning rule) has the form:

$$\begin{aligned} \boldsymbol{P}'_0 &= E[\|\boldsymbol{x}\|^2]\boldsymbol{I}_K, \ \boldsymbol{W}_0 = \boldsymbol{I}_K \\ \boldsymbol{y}_k &= \boldsymbol{W}_k \boldsymbol{x}_k \\ \boldsymbol{P}'_k &= \boldsymbol{P}'_{k-1} + \boldsymbol{y}_k \boldsymbol{y}_k^T \\ \Delta \boldsymbol{W}_k &= (\boldsymbol{P}'_k)^{-1} \boldsymbol{y}_k \left(\boldsymbol{x}_k - \boldsymbol{W}_k^T \boldsymbol{y}_k \right)^T \\ \boldsymbol{W}_{k+1} &= \boldsymbol{W}_k + \Delta \boldsymbol{W}_k \end{aligned} \tag{8}$$

where $\boldsymbol{I}_K$ is K-dimensional unit matrix.

If we replace $(\boldsymbol{P}'_k)^{-1}$ by the matrix $\boldsymbol{P}_k$ then it can be easily shown that the updating rule takes the form:

$$\Delta \boldsymbol{P}_k = -\frac{\boldsymbol{P}_{k-1} \boldsymbol{y}_k \boldsymbol{y}_k^T \boldsymbol{P}_{k-1}}{1 + \boldsymbol{y}_k^T \boldsymbol{P}_{k-1} \boldsymbol{y}_k} \tag{9}$$

Substituting $\boldsymbol{z} \doteq \boldsymbol{P}\boldsymbol{y}$ we get the final form of the improved PSA rule:

$$\begin{aligned} \boldsymbol{P}_0 &= E[\|\boldsymbol{x}\|^{-2}]\boldsymbol{I}_K, \ \boldsymbol{W}_0 = \boldsymbol{I}_K \\ \boldsymbol{y}_k &= \boldsymbol{W}_k\boldsymbol{x}_k \\ \boldsymbol{z}_k &= \boldsymbol{P}_{k-1}\boldsymbol{y}_k \\ \boldsymbol{P}_k &= \boldsymbol{P}_{k-1} - \boldsymbol{z}_k\boldsymbol{z}_k^T/(1+\boldsymbol{z}_k^T\boldsymbol{y}_k) \\ \Delta\boldsymbol{W}_k &= (\boldsymbol{P}_k\boldsymbol{y}_k)\left(\boldsymbol{x}_k^T - \boldsymbol{y}_k^T\boldsymbol{W}_k\right) \\ \boldsymbol{W}_{k+1} &= \boldsymbol{W}_k + \Delta\boldsymbol{W}_k \end{aligned} \tag{10}$$

Note that for $K = 1$ the improved RLS scheme (10) reduces to the RLS-Oja scheme (3). However, for image data when $K > 1$ many epochs are necessary to ensure the convergence (compare right plot in Fig. 1). We have noticed the deflation of data by the elimination of the signal projected on the principal axis, changes the situation dramatically: the improved RLS learning scheme converges in one epoch for the modified image data.

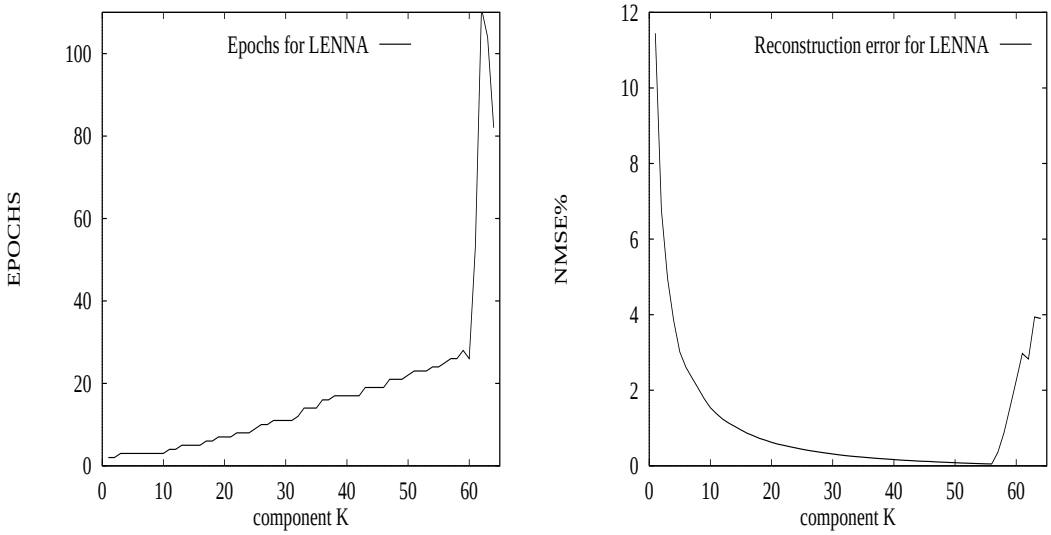

Fig. 1. *The convergence and quality of generalized RLS-Oja learning on the LENNA image as the function of the subspace dimension K.*

In order to avoid computation of the first principal vector necessary for the data deflation, we estimate the first principal vector by $\boldsymbol{u} \doteq \frac{1}{\sqrt{N}}\mathbf{1}$ where $\mathbf{1} = [1, \ldots, 1]^T$. In case of images for natural scenes, this estimation is very accurate. For instance, for the LENNA image, the correlation coefficient between $\boldsymbol{u}$ and the first principal vector is equal to 0.99999918577.

The deflation process of the vector $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ against the vector $\boldsymbol{u}$ which gives a vector $\tilde{\boldsymbol{x}}$ is algorithmically very simple and efficient:

$$\tilde{x}_i = x_i - \overline{x}, \quad \text{where} \quad \overline{x} = \frac{\left(\sum_{j=1}^N x_j\right)}{N}, \qquad i = 1, \ldots, N \ .$$

It is easy to prove that the time complexity of one learning step in the algorithm (10) is $O(KN + K^2)$, so the complexity of the whole algorithm working on

a M element vector data set is $O(MK(N+K))$=$O(MNK)$ as $K \leq N$. Hence the time complexity is bounded by a linear function of K what is confirmed by the experiments.

4. Improved PCA algorithms

The PSA algorithm discussed in previous sections learns the subspace spanned by the K principal components. In other words, such a fully symmetrical network can learn only a rotated basis of the principal component subspace. In order to extract true principal components, an asymmetry must be introduced in the learning algorithm ([10, 11, 2, 12]). Almost all known PCA algorithms related to our improved PSA algorithm can be presented in compact unified form as follows:

$$\begin{aligned}
\boldsymbol{P}_0 &= E[\|\boldsymbol{x}\|^{-2}]\boldsymbol{I}_K,\ \boldsymbol{W}_0 = \boldsymbol{I}_K \\
\tilde{\boldsymbol{x}}_k &= \boldsymbol{x}_k - \bar{x}\mathbf{1} \ \text{(optional deflation)} \\
\boldsymbol{y}_k &= \boldsymbol{W}_k\tilde{\boldsymbol{x}}_k \\
\boldsymbol{z}_k &= \boldsymbol{P}_{k-1}\boldsymbol{y}_k \\
\boldsymbol{P}_k &= \boldsymbol{P}_{k-1} - \boldsymbol{z}_k\boldsymbol{z}_k^T/(1+\boldsymbol{z}_k^T\boldsymbol{y}_k) \\
\Delta\boldsymbol{W}_k &= \boldsymbol{P}_k\left\{\boldsymbol{y}_k\tilde{\boldsymbol{x}}_k^T - \left[\Gamma .* \left(\boldsymbol{y}_k\boldsymbol{y}_k^T\right)\right]\boldsymbol{W}_k\right\} \\
\boldsymbol{W}_{k+1} &= \boldsymbol{W}_k + \Delta\boldsymbol{W}_k
\end{aligned} \tag{11}$$

where $\Gamma = [\gamma_{ij}]$ is $K \times K$ parametric matrix and $.*$ denotes component-wise multiplication of matrices.

By specifying properly the entries of matrix Γ we get the well known PCA algorithms. For instance:

1. Sanger's Generalized Hebbian Algorithm (GHA) [10]: $\gamma_{ij} = 1$ if $i \geq j$, $\gamma_{ij} = 0$ otherwise;
2. Oja's Stochastic Gradient Ascent (SGA) [2]: $\gamma_{ij} = 1$ if $i = j$, $\gamma_{ij} = 2$ if $i > j$, $\gamma_{ij} = 0$ otherwise;
3. Oja's-Ogawa-Wangiviwattana Weighted Subspace Algorithm (WSA) [2]: $\gamma_{ij} = i$ for $i = 1, \ldots, K$;
4. Brocket's Subspace Algorithm (BSA) [11]: $\gamma_{ij} = (K+1-j)/(K+1-i)$ for $i, j = 1, \ldots, K$;

Due to the limit of space our computer experiments results are limited only to the improved PSA algorithm (10).

5. Experimental results

We have conducted our experiments on several standard natural images which were converted into vector data sets by a block-wise scan.

K	NMSE% PCA	NMSE% PSA	12.5%	K	NMSE% PCA	NMSE% PSA	12.5%
1	6.120731	6.120731	6.124	9	0.329177	0.306260	0.297
2	3.192310	3.192250	3.104	10	0.235708	0.237839	0.235
3	1.975366	1.975362	1.885	11	0.166623	0.167247	0.167
4	1.428756	1.420785	1.308	12	0.129434	0.126868	0.132
5	0.902688	0.902725	0.895	13	0.096451	0.090180	0.088
6	0.721145	0.740387	0.692	14	0.056054	0.058058	0.059
7	0.577829	0.561724	0.525	15	0.024771	0.024546	0.024
8	0.421482	0.419243	0.414	16	0.000000	0.000000	0.000

Tab. 1. *Quality measures for PCA and PSA algorithms.*

In order to compare PSA solved by the improved RLS rule (10) with PCA computed by RLS-Oja rule (3), we calculate the percentage of signal energy (NMSE%) which is lost when we project the vector data onto a K-dimensional subspace. In Table 5. we show results for the LENNA image scanned by 4×4 blocks ($N = 16$). Columns denoted by 12.5% include the results of PSA when every 8-th vector from the data input is only taken for the training.

We see that the result of PCA and PSA techniques differ insignificantly for all components. However, the advantage of the PSA algorithm over the PCA based one (see [6, 12]) is that for PSA, the image data is presented only one time for $K \leq N/2$. For $K > N/2$ in order to provide the same level of accuracy for PSA, we have to repeat the deflation against the subspace spanned by the first $N/2$ determined basis vectors. However, for practical applications the accuracy of PSA which we get for $K < N/2$ is sufficient.

We also see from the column marked by 12.5% in the above table that our algorithm gives a good performance in case of incomplete data.

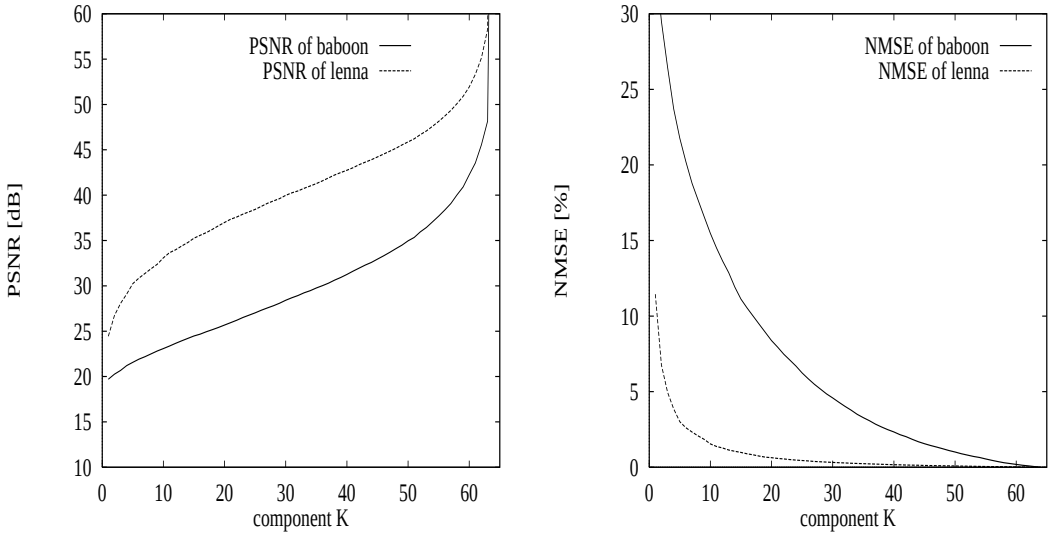

Fig. 2. *Quality measures for LENNA and BABOON as a function of the subspace dimension K.*

The two plots in Fig. 2 show the quality of the image reconstruction using K components for LENNA and BABOON scanned by 8×8 blocks ($N = 64$). This

quality is measured by the peak signal to noise ratio (PSNR) in decibels and by NMSE% measure.

Figure 3 illustrates reconstructed images obtained for $K = 3$ components ($N = 16$) with PCA and improved PSA approaches. They are indistinguishable what confirms coincidence of subspaces found by both approaches.

We conducted also an experiment when the image BABOON was reconstructed using the 3D subspace learned for LENNA image (right side). When it was reconstructed using 3D subspace learned on its own data (left side), the quality measured by PSNR improved only by 0.04[dB]. It means that the generalization property of the PSA network learned on LENNA is rather high.

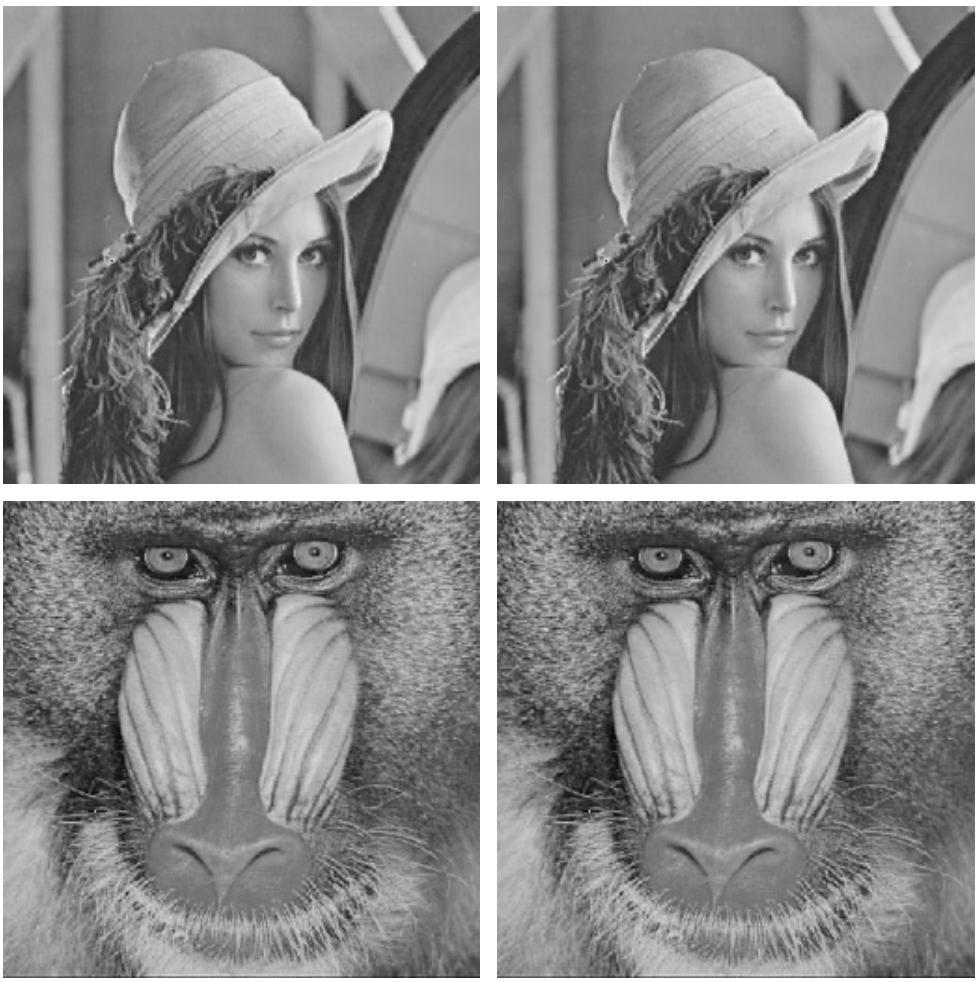

Fig. 3. *LENNA (left) – PCA for* $K = 3, N = 16$*, learned on LENNA, PSNR=32.53[dB]; LENNA (right) – PSA for* $K = 3, N = 16$*, learned on LENNA, PSNR=32.53[dB]; BABOON (left) – PSA for* $K = 3, N = 16$*, learned on BABOON, PSNR=23.04[dB]; BABOON (right) – PSA for* $K = 3, N = 16$*, learned on LENNA, PSNR=23.00[dB].*

6. Conclusions

Oja's learning scheme for the principal subspace analysis is refined by specification of RLS-based adaptation scheme for the learning rate matrix. It is experimentally

shown that an orthogonalization procedure performed as a preprocessing step on image data results in the convergence of the scheme in one epoch only. This speedup is very important for practical applications of PSA and PCA. Moreover, further speedup we can get by selecting only a small representative part of image data for training (about 10% is sufficient) since the proposed algorithm finds the subspaces with very good accuracy even on incomplete data of the image.

References

[1] Amari S., Neural theory of association and concept formation, Biological Cybernetics, 26, 1977, 175-185.

[2] Oja E., Ogawa H., Wangiviwattana J., Principal component analysis by homogeneous neural networks, Part I and II, IEICE Trans. on Information Systems (JAPAN), E75-D, 1992, 366-382.

[3] Cichocki A., Unbehanen R., Neural Networks for Optimization and Signal Processing, J. Wiley, 1994.

[4] Cichocki A., Unbehanen R., Robust estimation of principal components by using neural networks, Electronics Letters, 29, 1993, 1869-1870.

[5] Bannour S., Azimi-Sadjadi M.R., Principal component extraction using recursive least squares learning, IEEE Trans. Neural Networks, 6, 1995, 457-469.

[6] Cichocki A., Kasprzak W., Skarbek W., Adaptive learning algorithm for Principal Component Analysis with partial data, In: Proc. of the Thirteenth European Meeting on Cybernetics and Systems Research, Symposium on Artificial Neural Networks and Adaptive Systems, Vienna, 1996, (to appear).

[7] Skarbek W.: Methods for digital image representation, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1993 (in Polish).

[8] Oja E., Principal components, minor components, and linear neural networks, Neural Networks, 5, 1992, 927-935.

[9] Karhunen J., Stability of Oja's PCA subspace rule, Neural Computation, 6, 1994, 739-747.

[10] Sanger T. D., Optimal unsupervised learning in a single-layer feedforward neural network, Neural Networks, 2, 1989, 459-473.

[11] Brockett R.W., Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems, Linear Algebra Applications, 146, 1991, 79-91.

[12] Kasprzak W., Cichocki A., Recurrent least squares learning for quasi-parallel principal component analysis, In: Proc. European Symposium on Artificial Neural Networks ESANN'96, Belgium, 1996 (to appear).