Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 50 – 60

doi:10.3906/elk-1303-151

Noise adjusted version of generalized principal component analysis

Mojtaba AMINI OMAM, Farah TORKAMANI-AZAR*
Cognitive Communication Research Group, Faculty of Electrical and Computer Engineering,
Shahid Beheshti University, G.C., Evin, Tehran, Iran

Received: 21.03.2013 • **Accepted/Published Online:** 25.09.2013 • **Final Version:** 01.01.2016

Abstract: Principal component analysis (PCA) is a well-known tool in image processing, especially in dimension reduction schemes. Since using PCA is based on the vector representation of the image, the spatial locality of pixels in the image is not considered. However, generalized PCA (GPCA) could be applied to images in two dimensional spaces. Both schemes do not consider the noisy case of signals. Noise adjusted PCA (NAPCA) tries to find new coordinates for signal representation based on signal to noise ratio (SNR) maximization. In this paper we generalized noise adjusted GPCA to benefit the advantage of GPCA and SNR maximization case of NAPCA in two dimensional spaces. The experimental results on the huge databases show its reliability.

Key words: Principal component analysis, generalized principal component analysis, signal to noise ratio improvement, noise adjusted principal component analysis

1. Introduction

Principal component analysis (PCA) is mathematically defined as an orthogonal linear transformation that transforms a number of possibly correlated variables into a number of uncorrelated variables called the principal components [1]. To achieve this aim, first of all, the PCA theorem calculates the eigenvalue decomposition of a data covariance matrix or singular value decomposition (SVD) of a data matrix, usually after mean centering the data for each attribute. These eigenvalues are sorted decreasingly and then their corresponding eigenvectors computed. It means that the eigenvalues define the variance of any projection data on their eigenvectors as new coordinates.

To apply the PCA theorem on images, their vector based representation should be obtained, which leads to a loss of spatial locality information and requires a lot of time and space costs. In 2004, the generalized PCA (GPCA) algorithm was proposed to transform the image into a smaller matrix and to preserve the spatial information with lower computational costs than PCA [2,3]. In GPCA, k images as $\{A_1, A_2, \ldots, A_k\}$ in m × n size with a mean of zero are considered. The main aim is to compute two matrices, C in $m \times d_1$ size and R in n × d_2 size, with orthonormal columns such that $variance(C,R) = \sum_{j=1}^{k} \left\| C^T A_j R \right\|_F^2$ is maximized [3]. C and R are defined as eigenmatrices instead of eigenvectors. These matrices are obtained from the image rows and columns covariance matrix to project the image matrices on them. Now the projection of each image on

*Correspondence: f-torkamani@sbu.ac.ir

two matrices, C and R, is determined by Eq. (1):

$$D_j = C^T A_j R. \tag{1}$$

In this case D_j is in size $d_1\times\ d_2$. Therefore, instead of storing the k image matrix in size m $\times$ n, the k matrix is stored in size $d_1\times\ d_2$ so that the two principal matrices C and R could be done. Although PCA and GPCA project data on the dimensions that have the strongest variance in the dataset, these two algorithms do not consider the signal to noise ratio (SNR) in the noisy data case. Therefore, to increase SNR, Green et al. developed a maximum noise fraction (MNF) transform based on the maximization of SNR [4]. Later Lee et al. used a two-stage method comprising noise whitening and PCA processes [5]. In this new transform, called noise adjusted principal components analysis (NAPCA), principal components are ranked by SNR maximizing rather than by variance maximizing [5,6]. In the NAPCA observation model, k vectors $\{A_1 + n_1,\ A_2 + n_2,\ldots,\ A_k + n_k\}$ in size m $\times$ 1 with a mean of zero are considered, where noise, n_i, and signal, A_i, are independent. The covariance matrix is $\Sigma = \Sigma_a + \Sigma_n$, where Σ_a and Σ_n are the signal and noise covariance matrices, respectively. Applying PCA on Σ_n leads to Eq, (2):

$$E^T \sum_n E = \Delta_n, \tag{2}$$

where E and Δ_n are eigenvectors and eigenvalues matrices of the noise covariance matrix, respectively. It is noted that Δ_n is the diagonal matrix. Therefore, $F = E\Delta_n^{-1/2}$ is the noise whitening matrix that normalizes Σ_n as $F^T\Sigma_n F = I$, where I is the identity matrix. Next, Σ is transformed as Eq. (3) to compute a noise-adjusted data covariance matrix called Σ_{na} [6]. In fact, Eq. (3) removes the noise effect from Σ.

$$\sum_{na} = F^T \sum F = F^T \sum_a F + I \tag{3}$$

Applying PCA on $\sum_{na}$ results in:

$$U^T \sum_{na} U = \Delta_{na}, \tag{4}$$

where U and Δ_{na} are the eigenvectors and eigenvalues of matrix $\sum_{na}$, respectively. The desired NAPCA can be obtained by the projection matrix:

$$Q = UF \tag{5}$$

In this case, the projection of matrix A on matrix **Q** as Y = A Q = A U F could be defined. The rows of matrix **Q** are ordered as the provided maximum SNR in the new database, **Y**. Estimation of the noise covariance matrix from the data a priori is one of the major disadvantages of this approach [7]. However, problems such as time requirements, memory costs, and vector based representation of images are found in PCA as well as in NAPCA algorithms.

In this paper, we decide to propose the noise adjusted GPCA, which could be attractive in image processing applications. In this idea, spatial locality information and SNR maximizing could be preserved simultaneously. In Section 2, GPCA is introduced and the proposed noise adjusted GPCA is explained in the following section. The paper is finalized by some simulation results to compare GPCA and NAGPCA.

2. Mathematical basics of GPCA in two-dimensional spaces

In PCA, for an image database with k images in size m × n, we need three m n × m n matrices to represent the covariance and the eigenvector matrices and m n scalar for representing the eigenvalues. On the other hand, calculating these huge covariance and eigenvector matrices needs a high CPU power. The loss of pixel neighboring properties in PCA is another important issue that makes PCA an improper method for image database processing. So, some authors offered the generalized form of PCA as GPCA [3].

Let us consider k image matrices as A_j , for $j = 1, ..., k$, with a mean of zero in size $m\times$ n. In GPCA, the main aim is to compute two matrices C in size $m\times$ d_1 and R in size $n \times d_2$ with orthonormal columns such that Eq. (6) is maximized:

$$\sum_{j=1}^{k} \left\|C^T A_j R\right\|_F^2 = \sum_{j=1}^{k} trace\left(C^T A_j R R^T A_j^T C\right). \tag{6}$$

In this situation the KL transform is defined as the projection of each image on two matrices, C and R, by Eq. (7):

$$D_j = C^T A_j R, \text{ for } j = 1, 2, \cdots, k \tag{7}$$

C and R are obtained from eigenvectors of the image's weighted rows and weighted columns covariance matrix, MR and MC, as:

$$MC = \sum_{j=1}^{k} A_j R R^T A_j^T;$$

$$MR = \sum_{j=1}^{k} A_j^T C C^T A_j. \tag{8}$$

Similar to the KL transform based PCA where the first largest eigenvalues and corresponding eigenvectors could be used for the reconstruction of data, in KL transform based on GPCA the first d_1 columns of C (the first d_1 eigenvectors corresponding to the first largest eigenvalues of MC) and the first d_2 columns of R (the first d_2 eigenvectors corresponding to the first largest eigenvalues of MR) could be used to reconstruct a good estimation of data. In this situation D_j would be in size $d_1 \times d_2$ (Eq. (1)) [5]. Unfortunately, there is not a closed solution to find C and R and it should be solved using iteration algorithms; however, the solution is found only after a few iterations. Let i = 0 to start iterations, and $\mathbf{C}_0$ equals the identity matrix to initialize. The GPCA algorithm block diagram is given in Figure 1.

Experiments show that the algorithm converges only after two to four iterations. To store a dataset containing k matrix image in size m × n, the required memory is k × m × n space without any processing. However, using GPCA leads to storing a database in $k \times d_1 \times d_2 + m \times d_1 + n \times d_2$ memory space. For example, for $k = 10,000, m = n = 145$ and $d_1 = d_2 = d$; Table 1 shows the percentage of saving.

The computational load for this method as compared with PCA is so low because of the calculation of the smaller covariance and eigenvector matrices [3].

In the following section, noise adjusted GPCA, which could be attractive in image processing applications, is introduced. In this idea, spatial locality information and SNR maximizing could be preserved simultaneously.

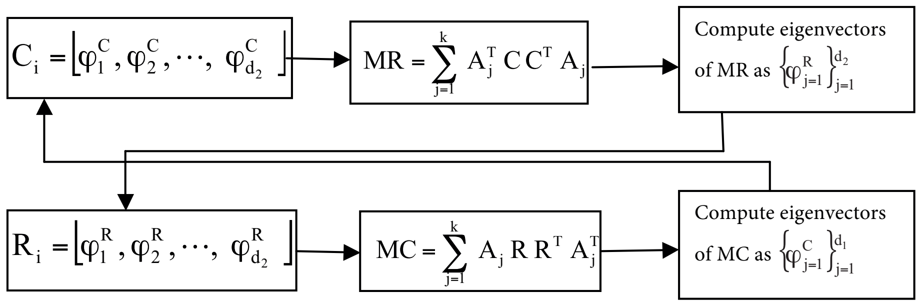


Figure 1. GPCA iteration algorithm block diagram.

Table 1. Saving percentages using GPCA (details are in the text).

d	50	75	100	110	120
Saving percentage	88%	73%	52%	42%	31%

3. The proposed idea

Since GPCA is a generalized form of PCA, a noise adjusted version of GPCA (NAGPCA) could be obtained. The aim of NAGPCA is to transfer images in size m × n to different matrices in size $d_1 \times d_2$ that are smaller than the original matrices to save memory and also increase SNR after recovering. Let us consider k noisy matrices as $A_j = S_j + N_j$, for j= 1, . . ., k, with a mean of zero. The GPCA row covariance matrix (MR) and column covariance matrix (MC) are obtained as Eqs. (9) and (10) (in the definition of NAGPCA, we used the symbols $\overline{C} and \overline{R}$):

$$MC = \sum_{j=1}^{k} A_j \bar{R}\bar{R}^T A_j^T = \sum_{j=1}^{k} S_j \bar{R}\bar{R}^T S_j^T + \sum_{j=1}^{k} N_j \bar{R}\bar{R}^T N_j^T \tag{9}$$

$$MR = \sum_{j=1}^{k} A_j \bar{C}^T \bar{C} A_j^T = \sum_{j=1}^{k} S_j \bar{C}^T \bar{C} S_j^T + \sum_{j=1}^{k} N_j \bar{C}^T \bar{C} N_j^T, \tag{10}$$

where $\overline{C}$ and $\overline{R}$ matrices should be computed to transform A_j matrices as defined in Eq. (7). In the noise adjusted version, we redefine the new forms for MR and MC matrices to reduce the noise effects as $\overline{MR} and \overline{MC}$. This process could be done using two whitening matrices F_C and F_R subjected to Eq. (11) and Eq. (12):

$$F_R^T \left(\sum_{j=1}^{k} N_j \bar{R}\bar{R}^T N_j^T \right) F_R = I \tag{11}$$

$$F_C^T \left(\sum_{j=1}^{k} N_j \bar{C}\bar{C}^T N_j^T \right) F_C = I \tag{12}$$

Therefore $\overline{MR} and \overline{MC}$ are obtained as Eq. (13) using Eqs. (9)–(12):

$$\overline{MC} = F_R^T \times MC \times F_R; \overline{MR} = F_C^T \times MR \times F_C. \tag{13}$$

Therefore, we introduce the transfer matrices $\overline{C}$ and $\overline{R}$, which contain d_1 and d_2 first eigenvectors of $\overline{MC} and \overline{MR}$ corresponding to d_1 and d_2 largest eigenvalues, respectively. Eq. (1) in this case is changed to Eq. (14) to transform each matrix A_j to a new smaller matrix as $\bar{D}_j$:

$$\bar{D}_j = \bar{C}^T A_j \bar{R}; j = 1, 2, \cdots, k. \tag{14}$$

The iterative algorithm for NAGPCA to find $\overline{C}$ and $\overline{R}$ is operated as GPCA. Its steps in summary are as follows:

Algorithm NAGPCA: (i is the iteration number)

1. $\overline{C}_1 \leftarrow [\mathrm{I}_{d1}, 0]^T$;
2. Compute $N_R = \sum_{j=1}^{k} N_j^T.\bar{C}_i.\bar{C}_i^T.N_j$;
3. Apply PCA to N_R as $V_{NR}^T.N_R.V_{NR} = \Delta_{NR}$ and find $F_{Ci} = V_{NR}.\Delta_{NR}^{-1/2}$;
4. Obtain $\overline{MR} = F_{Ci}^T \left[\sum_{j=1}^{k} A_j^T.\bar{C}_i.\bar{C}_i^T.A_j \right] F_{Ci}$ and compute the d_2 eigenvectors $\left\{ \phi_{j=1}^{\bar{R}} \right\}_{j=1}^{d_2}$ of $\overline{MR}$ corresponding to the d_2 largest eigenvalues of $\overline{MR}$;
5. Consider $\bar{R}_i = \left[\phi_1^{\bar{R}}, \phi_2^{\bar{R}}, \cdots, \phi_{d_2}^{\bar{R}} \right]$;
6. Compute $N_C = \sum_{j=1}^{k} N_j.\bar{R}_i.\bar{R}_i^T.N_j^T$
7. Apply PCA to N_C: $V_{Nc}^T.N_C.V_{NC} = \Delta_{NC}$, and find $F_{Ri} = V_{NC}.\Delta_{NC}^{-1/2}$;
8. Obtain $\overline{MC} = F_{Ri}^T \left[\sum_{j=1}^{k} A_j.\bar{R}_i.\bar{R}_i^T.A_j^T \right] F_{Ri}$ and compute the first d_1 eigenvectors $\left\{ \phi_{j=1}^{\bar{C}} \right\}_{j=1}^{d_1}$ of $\overline{MC}$ corresponding to the d_1 largest eigenvalues of $\overline{MC}$;
9. Consider $\bar{C}_i = \left[\phi_1^{\bar{C}}, \phi_2^{\bar{C}}, \cdots, \phi_{d_2}^{\bar{C}} \right]$;
10. Compute the transfer images in ith iteration as: $\bar{D}_{ji} = \bar{C}_i^T.A_j.\bar{R}_i$, for j=1, . . .,k;
11. Compute the reconstructed images in ith iteration as: $\bar{A}_{ji} = C_i.\bar{D}_{ji}.R_i^T$, for j=1, . . .,k;
12. Compute the error between the original images, A_j, and the reconstructed images in ith iteration, $\bar{A}_{ji}$, as $RMSE(i) = \sqrt{\frac{1}{k} \sum_{j=1}^{k} \left\| A_j - \bar{A}_{ji} \right\|_F^2}$;
13. If $RMSE(i) - RMSE(i-1) \leq thresholdlevel$, stop and go to step 14, otherwise go to step 2;

14. Set the transferred matrices and the transferred images as: $\bar{C} = \bar{C}_i, \bar{R} = \bar{R}_i, \bar{D}_j = \bar{C}^T.A_j.\bar{R} forj = 1, \cdots, k.\square$

To recover matrices A_j, as $\bar{A}_j$, Eq. (15) is used:

$$\bar{A}_j = \bar{C}.\bar{D}_j.\bar{R}^T forj = 1, \cdots, k \tag{15}$$

So, instead of storing A_j matrices, $\bar{D}_j$ matrices, $\bar{C}$, and $\bar{R}$ should be stored. The complexity of NAGPCA is comparable with GPCA and only the computations of N_R and N_C are added. It should be noted that with small d_1 and d_2, we lose a lot of information, so the SNR would be very small. By increasing d_1 and d_2, more information is available after reconstruction, and since the eigenvector orientation maximizes the SNR, therefore, the SNR will be increased. It should be mentioned that by increasing d_1 and d_2, in fact we reconstruct the entire set of signals, which also contains noise, so therefore the SNR decreases and the SNR curves of GPCA and NAGPCA approach each other. However, in all cases, NAGPCA had better performance than GPCA. As mentioned, in NAGPCA as well as in NAPCA, the biggest problem is estimation of the noise level. Because in our work noise covariance estimation is not the original intention, in our experiment the noise sample was assumed as given. However, the procedure could be repeated with different amounts of noise samples to get the best results.

4. Experimental results and discussion

In this section the effectiveness of the NAGPCA method in the noisy dataset reconstruction is shown. For our experiments two types of datasets were used. The first database was ORL faces that contained 400 images in size of 112 × 92 (available at http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.tar.Z). There were ten different images of each of the 40 distinct subjects. For some subjects, the images were taken at different times, in varying light, facial expressions (open/closed eyes, smiling/not smiling), and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Several image samples of the ORL database are shown in Figure 2a.

The second dataset employed the airborne visible infrared imaging spectrometer (AVIRIS). This hyperspectral image database has 220 bands ranging from 0.4 μm to 2.5 μm wavelength (available at ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/). Each image is of a 145 × 145 pixel size. Several image samples of the AVIRIS database are shown in Figure 2b. These datasets have a very important difference. The images of AVRIS are taken from a specific area with fixed objects as shown in Figure 2b. So, to store and reconstruct, the AVRIS dataset requires a small number of eigenvectors. However, the images of the ORL dataset contain 400 persons with different poses as shown in Figure 2a, and to store and reconstruct them, a large number of eigenvectors is required. This fact is also confirmed by Figure 2c, which shows the normalized eigenvalues of the datasets using ordinary PCA factorization.

First, each dataset was corrupted with white Gaussian noise of different variances. We assumed that the noise information is not available. So, we only estimated the noise variance using MATLAB command (randn(size(database))*estimated noise variance)) and constructed matrices NC and NR of steps 2 and 6 of the proposed algorithm. Each noisy dataset was used as an input to both algorithms, GPCA and NAGPCA. Experiments were done in three noisy cases with different reduced dimension values. Comparisons between the original images (without noise) in the two datasets and the reconstructed images in the three noisy cases and

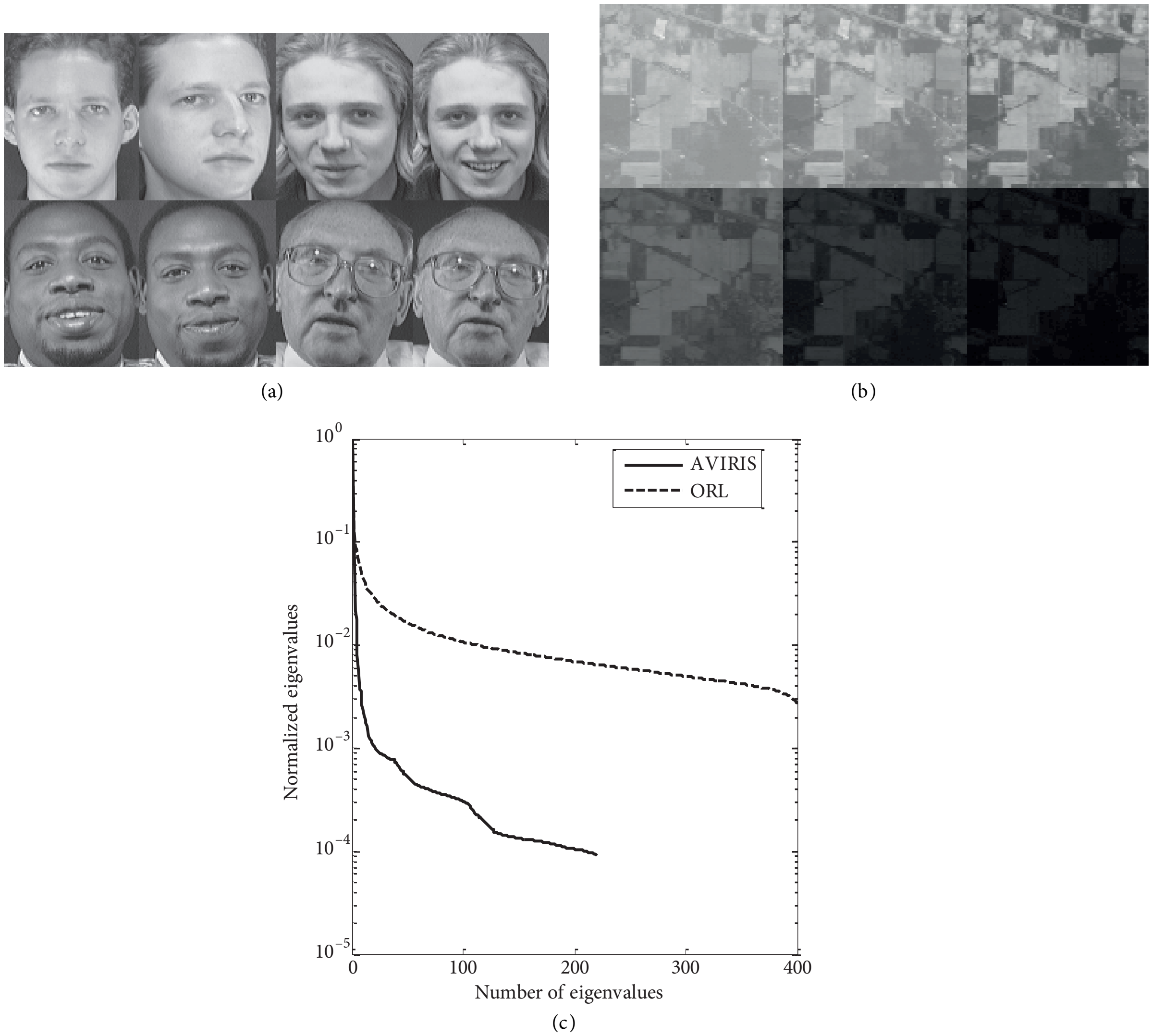


Figure 2. Several image samples from two databases: a) AVIRIS data; b) ORL face database; c) normalized eigenvalues of AVIRIS and PIX datasets.

different dimensions were completed with computation of the peak signal to noise ratio (PSNR) and one image quality measurement as SSIM. The structural similarity (SSIM) index is a method for measuring the similarity between two images [8]. The SSIM index is a full reference metric; in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as a reference. In this research, the database of ORL was corrupted using white Gaussian noise with different noise variances to acquire 400 noisy images with SNR = 18 dB, 25 dB, and 32 dB with respect to the original ORL images. Since these images were in the 112 × 92 size, $d_1 \in \{30, 40, 50, 60, 70, 80, 90, 100, 110\}$ and d_2 = round ($d_1 \times$ 92 / 112) were used in the simulation. The PSNR and SSIM of the entire noisy database with respect to the entire original database are shown in Figure 3 (with SNR = 18 dB in Figure 3a, 25 dB in Figure 3b, and 32 dB in Figure 3c).

The AVIRIS images were also corrupted using white Gaussian noise with different noise variances to attain noisy images with SNR = 32 dB, 25 dB, and 18 dB with respect to the original AVIRIS database. Since these images are in the 145 × 145 size, $d_1 = d_2 = d \in \{40, 50, 60, 70, 80, 90, 100110, 120, 130, 140\}$ were used in the simulation. The PSNR and SSIM of the entire noisy database with respect to the entire original database are shown in Figure 4 (with SNR = 18 dB in Figure 4a, 25 dB in Figure 4b, and 32 dB in Figure 4c).

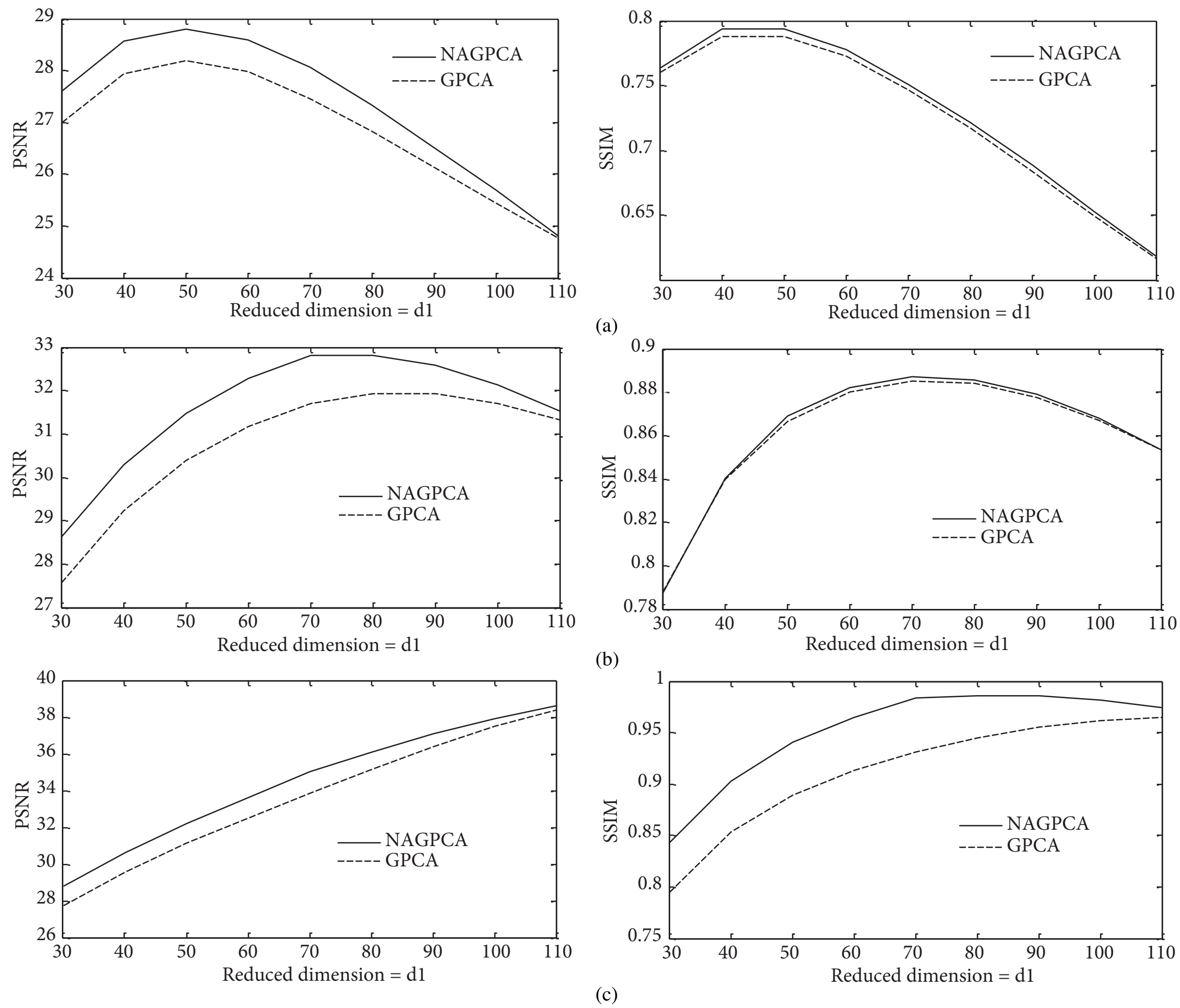


Figure 3. Simulation results with different noisy ORL datasets: a) added noise variance = 15 with a noise input dataset SNR = 18 dB; b) added noise variance = 7 with a noise input dataset SNR = 25 dB; and c) added noise variance = 3 with a noise input dataset SNR = 32 dB.

As shown in Figures 3 and 4, in the case of low noise (Figures 3c and Figure 4c) NAGPCA and GPCA have the same results and the NAGPCA algorithm can always be used, even in the case without noise. However, in the cases of higher noise, the NAGPCA algorithm leads to better results. In all plots, there is approximately an optimum point for which high PSNR and SSIM are obtained. In the PCA theorem, the first eigenvalues (corresponding eigenvectors) refer to the low frequency components and the last eigenvalues (corresponding eigenvectors) refer to the high frequency components, which in the noisy case show the noise effects. As mentioned in the previous section, since by using a large value of d the algorithm in fact uses noise components, and PSNR converges or decreases. In addition, the optimum point, or the optimum d for dimension reduction with increasing noise, decreased.

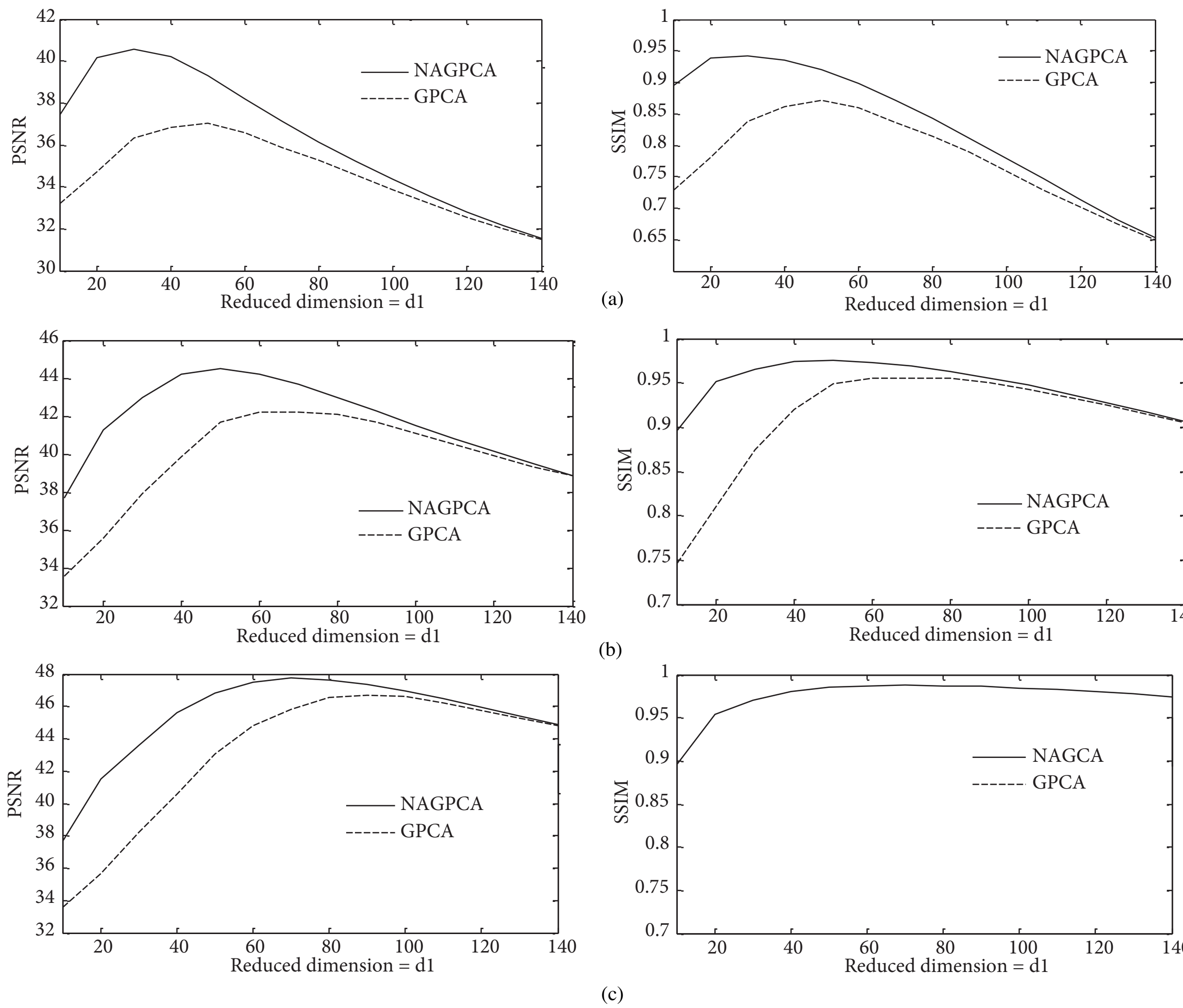
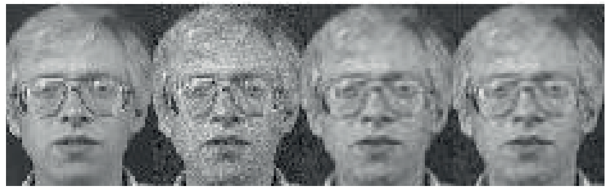


Figure 4. Simulation results with different noise AVIRIS datasets: a) added noise variance = 15 with a noise input dataset SNR = 32 dB; b) added noise variance = 7 with a noise input dataset SNR = 39 dB; and c) added noise variance = 3 with a noise input dataset SNR = 44 dB.

In the high noise case, the last eigenvalues (corresponding eigenvectors) contain noise, so the optimum point was prepared with a small d. However, without noise, we tend to a normal case, the number of eigenvectors used increases, and the quality improves. Therefore, in the low noise case, PSNR and SSIM increase approximately monotonically (Figures 3c and 4c). Figures 5a, 5b, and 5c show some samples of the original database, the noise database, and the reconstructed images in the optimum state of high, middle, and low noise cases of the ORL database. Figures 6a, 6b, and 6c show some samples of the original database, the noise database, and the reconstructed images in the optimum state of high, middle, and low noise cases of the AVIRIS database.

Before the optimum point, the d value is too small to reconstruct images with their suitable energy. After the optimum point, the d value is too large, which is similar to considering all available signals plus noise that leads to decreased PSNR in the noise cases. The percentages of saving in all cases are shown in Tables 2 and 3. The quality of the reconstructed images should be judged by looking at their percentages of saving.



(a)

(b)

(c)

Figure 5. Left to right: a sample of the original ORL database, a sample of the noise database, a sample of the database reconstructed by GPCA in optimum point, a sample of the database reconstructed by NAGPCA in optimum point. a) Data noise with SNR = 18 dB, optimum point $d_1 = 50$, $d_2 = 41$, saving percentage = 79%; b) data noise with SNR = 25 dB, optimum point $d_1 = 70$, $d_2 = 57$, saving percentage = 60%; c) data noise with SNR = 32 dB, optimum point $d_1 = 110$, $d_2 = 90$, saving percentage = 3%.

Table 2. Saving percentages in the ORL database.

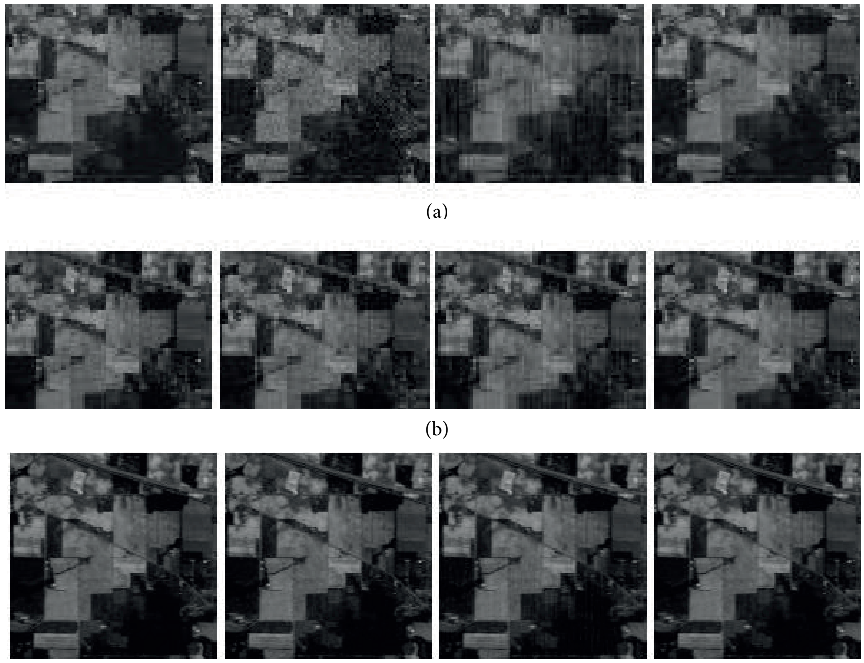
d_1	30	40	50	60	70	80	90	100	110
d_2	25	33	41	49	57	65	73	82	90
Saving percentage	92%	87%	79%	71%	60%	48%	35%	19%	3%

Table 3. Saving percentage in the AVIRIS database.

d = $d_1 = d_2$	10	20	30	40	50	60	70	80	90	100	110	120	130	140
Saving percentage	99%	97%	95%	91%	87%	81%	75%	68%	60%	50%	40%	29%	17%	4%

5. Conclusion

In this paper an algorithm called NAGPCA based on the same procedure that is used in NAPCA was proposed. This algorithm uses the advantage of GPCA, which is matrix based, instead of the vector based PCA. It simultaneously uses eigenimages ordered with respect to maximizing SNR. The required time and the computation complexity of the NAGPCA differ from the GPCA only by the calculation of two noise matrices. Experimental results show that by using the NAGPCA algorithm in the noise image databases, higher reconstruction performance is obtained as compared with the GPCA algorithm. Also, in the noiseless case, the implementation of



(a) (b) (c)

Figure 6. Left to right: a sample of the original VIRIS database, a sample of the noise database, a sample of the database reconstructed by GPCA in optimum point, a sample of the database reconstructed by NAGPCA in optimum point. a) Data noise with SNR = 18 dB, optimum point $d_1 = 30$, saving percentage = 95%; b) data noise with SNR = 25 dB, optimum point $d_1 = 50$, saving percentage = 87%; c) data noise with SNR = 32 dB, optimum point $d_1 = 70$, saving percentage = 75%.

the NAGPCA is the same as the GPCA. So, the NAGPCA could be used in all cases. In the NAGPCA, similar to the NAPCA, identifying the noise sample is done with approximate estimation.

References

[1] Jolliffe IT. Principal Component Analysis. 2nd ed. New York, NY, USA: Springer, 2002.

[2] Vidal R, Ma Y, Sastry S. Generalized principal component analysis (GPCA). IEEE T Pattern Anal 2005; 27: 1–15.

[3] Jieping Y, Ravi J, Qi L. GPCA: An efficient dimension reduction scheme for image compression and retrieval. In: 2004 Knowledge Discovery and Data Mining Conference; 22–25 August 2004; Seattle, WA, USA: SIGKDD'04, pp. 354–363.

[4] Green A, Berman M, Switzer P, Craig MD. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. IEEE T Geosci Remote 1988; 26: 65–74.

[5] Lee JB, Woodyatt AS, Berman M. Enhancement of high spectral resolution remote sensing data by a noise-adjusted principal components transform. IEEE T Geosci Remote 1990; 28: 295–304.

[6] Roger RE. A fast way to compute the noise-adjusted principal components transform matrix. IEEE T Geosci Remote 1994; 32: 1194–1196.

[7] Chang CI, Du Q. Interference and noise-adjusted principal components analysis. IEEE T Geosci Remote 1999; 37: 2387–2396.

[8] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE T Image Process 2004; 13: 600–612.