Learning Schemes in Using PCA Neural Networks for Image Restoration Purposes

ION ROSCA
Department of Computer Science
Academy of Economic Studies
Piata Romana # 6, Bucharest 1
ROMANIA
rosca@ase.ro

LUMINITA STATE
Department of Computer Science
University of Pitesti
Targu din Vale #1, Pitesti
ROMANIA
radus@sunu.rnc.ro

CATALINA LUCIA COCIANU
Department of Computer Science
Academy of Economic Studies
Piata Romana # 6, Bucharest 1
ccocianu@ase.ro

Abstract:- Image restoration methods are used to improve the appearance of an image by application of a restoration process that uses a mathematical model for image degradation. The restoration can be viewed as a process that attempts to reconstruct or recover an image that has been degraded by using some a priori knowledge about the degradation phenomenon. Principal component analysis allows the identification of a linear transformation such that the axes of the resulted coordinate system correspond to the largest variability of the investigated signal. The advantages of using principal components reside from the fact that bands are uncorrelated and no information contained in one band can be predicted by the knowledge of the other bands, therefore the information contained by each band is maximum for the whole set of bits. The multiresolution support set is a data structure suitable for developing noise removal algorithms. The multiresolution algorithms perform the restoration tasks by combining, at each resolution level, according to a certain rule, the pixels of a binary support image. The multiresolution support can be computed using the statistically significant wavelet coefficients. We investigate the comparative performance of different PCA algorithms derived from Hebbian learning, lateral interaction algorithms and gradient–based learning for digital signal compression and image processing purposes. The final sections of the paper focus on PCA based approaches for image restoration tasks based on the multirezolution support set as well as on PCA based shrinkage technique for noise removal. The proposed algorithms were tested and some of the results are presented and commented in the final part of each section.

Key-Words:- dimensionality reduction, image restoration, shrinkage functions, neural networks

1 Introduction

Images acquired from practical sources suffer from various degradation which may distort the representative features of the scene and make visualization and image analysis very difficult. The degradation mechanism is closely related to the physical process involved, typical examples include finite resolution of sensor arrays, motion blur, refraction, poor focus, and noise effects due to quantization and data transmission.

Noise is any undesired information that contaminates an image and appears in images from a variety of sources. In image processing, using statistical generative models enables principle derivation of methods for denoising, compression and other operations.

Typically, the noise can be modeled by Gaussian, uniform or salt and pepper distribution. The Gaussian model is most often used to model natural noise processes, such as those occurring from electronic noise in the image acquisition system. The salt and pepper type noise is typically caused by malfunctioning pixel elements in the camera sensors, faulty memory locations, or timing errors in the digitization process.

Because both the uniform and Gaussian noises provide unbiased or neutral noise model they are often used as a plausible noise model especially in evaluating the performance and comparative analysis of various denoising algorithms.

Several neural network-based approaches were proposed for extracting the principal components of a stationary vector stochastic process directly from the input data, the research efforts being oriented mainly toward to new neural network designs, to extend the existing neural network models for feature extraction and data projection and respectively to establish links between classical approaches and neural networks.

An important feature of neural networks is the ability they have to learn from their environment, and, through learning to improve performance in some sense.

One of the new trends is the development of specialized neural architectures together with classes of learning algorithms to provide alternative tools for solving feature extraction and data projection problems.

In the following we restrict the development to the problem of feature extracting unsupervised neural networks derived on the basis of the biologically motivated Hebbian self-organizing principle which is conjectured to govern the natural neural assemblies and the classical principal component analysis (PCA) method used by statisticians for almost a century for multivariate data analysis and feature extraction.

We investigate the comparative performance of different PCA algorithms derived from Hebbian learning, lateral interaction algorithms and gradient–based learning for digital signal compression and image processing purposes.

Aiming to obtain a guideline for choosing a proper method for a specific application we developed a series of simulations on some the most currently used PCA algorithms as GHA, Sanger variant of GHA and APEX.

The paper reports the conclusions experimentally derived on the convergence rates and their corresponding efficiency for specific image processing tasks.

The final sections of the paper focus on PCA based approaches for image restoration task and on PCA based shrinkage technique for noise removal.

The research work reported in the paper aims to propose a new image reconstruction method based on the features extracted from the noise given by the principal components of the noise covariance matrix. The computation of the features of the noise $\eta \sim N(0,\Sigma)$ is carried out by a PCA neural network trained by the GHA as well as alternative approaches as for instance Generalized Recursive Least Square (GRLS) algorithm and APEX . The training process aims the learning of the p most significant eigen-vectors of Σ, that is the eigenvectors corresponding to the largest p eigen-values of Σ.

The restoration can be viewed as a process that attempts to reconstruct or recover an image that has been degraded by using some *a priori* knowledge about the degradation phenomenon. Thus restoration techniques are oriented toward modeling the degradation and applying the inverse process in order to recover the original image. This approach usually involves formulating a criterion of goodness that will yield some optimal estimate of the desired result.

The inverse-filter method works only for extremely high signal-to-noise-ratio images. The Wiener filter is usually implemented only after the WSS assumption has been made for images. Furthermore, knowledge of the power spectrum or correlation matrix of the clean image is required. Usually, additional assumptions regarding boundary conditions are made so that fast orthogonal transforms can be used. Approaches based on noncausal models such as the noncausal autoregressive of Gaussian-Markov random-field models also make assumptions such as WSS and periodic boundary conditions.

The image restoration tasks mainly correspond to the process of finding an approximation to the overall degradation process and finding the appropriate inverse process to estimate the original unknown image. Principal component analysis allows the identification of a linear transformation such that the axes of the resulted coordinate system correspond to the largest variability of the investigated signal. The signal features corresponding to the new coordinate system are uncorrelated, that is in case of normal models, these components are independent. The research reported in the paper aims the development of a methodology for noise removal in image restoration. The noise removal technique is based on a transform that decorrelates the components of the noise followed by the use of a shrinkage process in order to remove the noise.

2 Hebbian Learning in Training PCA System

In our approach, the input signal is modeled as a wide-sense- stationary n-dimensional process $\left(X(t), t \geq 0\right)$ of mean $E(X(t))=0$ and covariance matrix $E\left(X(t)X(t)^T\right)=S$. We denote by $\Phi_1,...,\Phi_n$ a set of orthonormal eigen-vectors of S taken according to the decreasing order of their corresponding eigen-values $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$.

Being given that the data dimensionality n can be very large, a sort of compression to allow representations of less dimensionality is desired. Let $\Psi_1,...,\Psi_n$ be an orthonormal base of $\mathbf{R}^{\mathbf{n}}$. Then $X(t)$ can be written without error

$$X(t)=\sum_{i=1}^{n} y_i(t)\Psi_i$$

$$y_i(t)=\Psi_i^T X(t)$$

Conventionally, $\Psi_1,...,\Psi_n$ are referred as features and $y_1(t),...,y_n(t)$ are called the values of these features at $X(t)$. Let m be the desired size of the representation. Then, a m-dimensional representation can be obtain by retaining the values of the features corresponding to a subset of size m of $\Psi_1,...,\Psi_n$. Assume without loss of generality that the selected features are $\Psi_1,...,\Psi_m$. Then we get $\hat{X}(t)$ an approximation of $X(t)$,

$$\hat{X}(t)=\sum_{i=1}^{m} y_i(t)\Psi_i$$

The square mean error is,

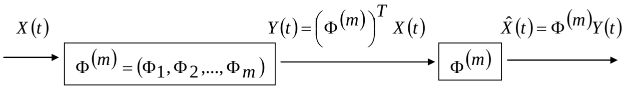
$$\varepsilon_m^2 = E\left(\left\|X(t)-\hat{X}(t)\right\|^2\right)$$

For given $\Psi_1,...,\Psi_n$, the best subset of m features is a subset that minimizes ε_m^2. Accordingly, the best representation corresponds to an orthonormal basis that minimizes ε_m^2.

From the point of view of the classical LMS criterion, the most informative directions of the process $\left(X(t), t \geq 0\right)$ are given by $\Phi_1,...,\Phi_n$ and accordingly, for any $m, 1 \leq m \leq n$ its LMS-optimal linear features are $\Phi_1,...,\Phi_m$.

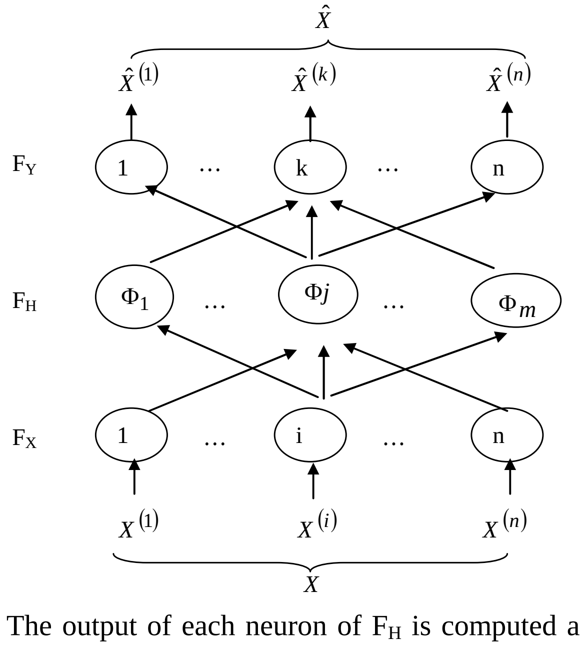
The LMS-optimal compression/decompression scheme is represented below,

$$1 \leq m \leq n$$


$X(t)$
$\Phi^{(m)}=(\Phi_1,\Phi_2,...,\Phi_m)$
$Y(t)=\left(\Phi^{(m)}\right)^T X(t)$
$\Phi^{(m)}$
$\hat{X}(t)=\Phi^{(m)}Y(t)$

$$\varepsilon_m^2 = E\left(\left\|X(t)-\hat{X}(t)\right\|^2\right)=\sum_{i=m+1}^{n}\lambda_i$$

The architecture of a PCA neural network consists of the n-neuron input layer and the m-neuron computation layer.



The output of each neuron of F_H is computed as a weighted linear combination of its inputs,

$$h_j(X)=\sum_{i=1}^{n} X^{(i)}\Phi_j^{(i)}=\Phi_j^T X = y_j$$

$h(X) = (h_1(X),...,h_m(X))^T$

Similarly, the output of each neuron of F$_Y$ is computed as a weighted combination of its inputs, that is,

$$\hat{X} = c_k(X) = \sum_{p=1}^{m} h_p(X)\Phi_j^{(p)} =$$

$$= (h(X))^T \Phi_j$$

The aim is to develop an adaptive learning algorithm to encode asymptotically $\Phi_1,...,\Phi_m$ as values of the synaptic vectors $W_1,...,W_m$ of the neurons in the computation hidden layer.

Let us denote by $W(t) = (W_1(t),...,W_m(t))$ the synaptic memory at the moment *t*, and let $Y(t) = (Y_1(t),...,Y_m(t))^T$ be the output of the computation layer, where $Y_j(t) = W_j^T(t)X(t), 1 \le j \le m$.

The Hebbian learning rule for learning the first principal component is,

$$W_1(k+1) = W_1(k) + \eta(k)X(k)Y_1(k) \qquad (2.1)$$

where the sequence of learning rates $(\eta(k))$ are taken such that the conditions of the Kushner theorem hold [8], $\sum_{k=1}^{\infty}\eta(k) = \infty$, $\lim_{k\to\infty}\eta(k) = 0$, there exists $p > 1$ such that $\sum_{k=1}^{\infty}(\eta(k))^p < \infty$.

The Equation (2.1) is not interesting because the resulting algorithm is unstable that is $\lim_{k\to\infty}\|W_1(k)\| = \infty$ but asymptotically $W_1(k)$ is directed toward $L(\Phi_1)$ the linear subspace generated by Φ_1 , hence the instability is the only obstacle inhibiting Equation (2.1) from being a real principal component analyzer. Consequently, the normalized version of the Hebbian learning rule

$$W_1(k+1) = \frac{W_1(k) + \eta(k)X(k)Y_1(k)}{\|W_1(k) + \eta(k)X(k)Y_1(k)\|}$$

(2.2) is a possible learning scheme for the first principal component but it is not suitable enough because it is not a local rule.

Oja and Karhunen [6] proposed the linearized version of (2.2) using a first order approximation scheme, the resulted learning scheme being known as the Oja's learning algorithm,

$$W_1(k+1) = W_1(k) + \eta(k)\left(X(k)Y_1(k) - Y_1^2(k)W_1(k)\right) \qquad (2.3)$$

The Generalized Hebbian Algorithm (GHA) [9] is one of the first neural models for extracting multiple PCs. The idea of GHA is to use the Hotelling deflation technique and the Oja's algorithm for learning as many principal components as are required. The GHA learning scheme is,

$$W_1(k+1) = W_1(k) + \eta(k)\left(X(k)Y_1(k) - Y_1{}^2(k)W_1(k)\right)$$

(2.4)

$$W_j(k+1) = W_j(k) + \eta(k)\left(\tilde{X}_j(k)\tilde{Y}_j(k) - \tilde{Y}_j{}^2(k)W_j(k)\right),$$

(2.5)

for $2 \le j \le m$, where

$$Y_j(k) = W_j^T(k)X(k),$$

$$\tilde{X}_j(k) = \tilde{X}_{j-1}(k) - Y_{j-1}(k)W_{j-1}(k) =$$

$$= \sum_{i=1}^{j-1} Y_i(k)W_i(k),$$

$$\tilde{Y}_j(k) = W_j^T(k)\tilde{X}_j(k).$$

The difficulty related to the GHA learning scheme arises from the fact that each neuron *j* of the system computes two outputs, $Y_j(k)$ and $\tilde{Y}_j(k)$, corresponding to the input in the system $X(k)$ and respectively to the "deflated" input $\tilde{X}_j(k)$. Each neuron is influenced by all neurons of smaller range, these influences establishing a sort of deflation process applied to the input. The output $\tilde{Y}_j(k)$ together with the deflated input $\tilde{X}_j(k)$ is used by the learning scheme of Oja type, while the output $Y_j(k)$ is used to compute the next step of deflation process, that is the deflated input of the (*j*+1)-th neuron.

The variant proposed by Sanger [16] simplifies the learning process by using only output of each neuron in both, the synaptic learning scheme and the input deflation. The Sanger variant of GHA is,

$$W_1(k+1) = W_1(k) + \eta(k)\left(X(k)Y_1(k) - Y_1{}^2(k)W_1(k)\right)$$

(2.6)

$$W_j(k+1) = W_j(k) + \eta(k)\left(\tilde{X}_j(k)Y_j(k) - Y_j{}^2(k)W_j(k)\right)$$

(2.7)

for $2 \le j \le m$, where

$Y_j(k) = W_j^T(k)X(k)$ and

$\tilde{X}_j(k) = \tilde{X}_{j-1}(k) - Y_{j-1}(k)W_{j-1}(k)$ is the input deflated at the level of the *j*th neuron.

Obviously, the synaptic learning algorithm can be also written as,

$$W_j(k+1)=W_j(k)+\eta(k)\left(X(k)Y_j(k)-Y_j(k)\sum_{i=1}^{j}Y_i(k)W_i(k)\right)$$

$$1\le j\le m \qquad (2.8)$$

The ODE assigned by the Kushner theorem is,

$$\frac{dW_1(t)}{dt}=SW_1(t)-\left(W_1^T(t)SW_1(t)\right)W_1(t), \qquad (2.9)$$

$$\frac{dW_j(t)}{dt}=SW_j(t)-\left(W_j^T(t)SW_j(t)\right)W_j(t)-$$

$$\sum_{i=1}^{j-1}\left(W_i^T(t)SW_j(t)\right)W_i(t),\ 2\le j\le m. \qquad (2.10)$$

The Adaptive Principal Component Extraction (APEX) rule was proposed by Kung and Diamantaras [7]. The model extracts multiple PCs using lateral connections between the output neurons instead of using an explicit, off-line deflation transformation.

The lateral connection from the *p*th neuron to the *j*th neuron is weighted by $a_{pj}(t)$, $2\le j\le m$, $1\le p\le j-1$.The learning equations changes the lateral connection weights and the synaptic memories according to,

$$W_j(k+1)=W_j(k)+\eta(k)\left(\tilde{X}_j(k)Y_j(k)-Y_j^{\,2}(k)W_j(k)\right)$$

(2.11)

$$Y_j(k)=W_j^T(k)X(k)-\sum_{i=1}^{j-1}a_{ij}(k)Y_i(k) \qquad (2.12)$$

Assuming that at the moment t_0, $W_i(t_0)\approx\Phi_i$, for all $1\le i\le j-1$, the ODE assigned to the APEX algorithm for encoding the *j*th eigen vector as the synaptic memory, according to Kushner theorem is,

$$\frac{dW_j(t)}{dt}=SW_j(t)-\sum_{l=1}^{j-1}\lambda_l\Phi_l a_{lj}(t)-\sigma_j(t)W_j(t), (2.13)$$

$$1\le j\le m$$

$$\frac{da_{ij}(t)}{dt}=\lambda_i\left(W_j^{\,T}(t)\Phi_i-a_{lj}(t)\right)-\sigma_j(t)a_{lj}(t), \quad (2.14)$$

$2\le j\le m$, $1\le i\le j-1$, where,

$$q_j(t)=W_j(t)-\sum_{l=1}^{j-1}a_{lj}(t)\Phi_l\ ,$$

$$\sigma_j(t)=q_j^T(t)Sq_j(t).$$

Note that the theoretical analysis [2], [5], [9], establishes the almost sure convergence of the above mentioned algorithms to the principal components in case the *m* largest eigen values of *S* are such that $\lambda_1>\lambda_2>\cdots>\lambda_m>\lambda_{m+1}\ge\cdots\ge\lambda_n$.

3 Tests Using PCA Systems in Solving Image Dimensionality Reduction Task

Let $I(t)$ be a wide-sense- stationary *N*-dimensional process of mean $E(I(t))$ resulted by sampling a given image I; $\bar{I}(t)=I(t)-E(I(t))$. Each sampled matrix $\bar{I}(t)$ is processed row by row, each row being split in lists of 15 consecutive components. We denote by $X(t)$ such a sublist and we assume that $X(t)\sim N(0,\Sigma)$.

In the following we denote by,

- $n=15$ - the dimension of $X(t)$;
- $m=3$ - the number of desired principal components;
- $t_{max}\in\{10,20,50,75\}$ - the number of the distorted variants of the image *I*;
- $\eta(t)=\dfrac{1}{(t+n)ln(t+n)}$ - the sequence of learning rates taken to satisfy the constrains considered in the Kushner theorem
- $W_0\in M_{15x3}(R)$ - the initial synaptic memories whose entries are randomly generated.

In case of the APEX algorithm, the initial values of the lateral connection weights are $\forall\, i\ge j, a_{ij}=0$ and for all $1\le i<j\le 3$, a_{ij} are randomly generated according to the uniform distribution on $[0,0.1)$.

The experiments were performed for several covariance matrices, the reported results being obtained with respect to the following examples,

$$\Sigma_1=diag\{15,10,5,0.1,0.1,...,0.1\}$$

$$\Sigma_2=diag\{10,6,2,0.04,0.04,...,0.04\}$$

$$\Sigma_3=diag\{4,2,1,0.01,0.01,...,0.01\}$$

In order to compare the convergence rates of the previously mentioned algorithms, for each sample, we computed,

- the empirical mean variation of the synaptic vectors on the final iteration,

$$V=\frac{1}{m}\sum_{i=1}^{m}D\left(W_i(t_{\max}),W_i(t_{\max}-1)\right),$$

$$D\left(W_i\left(t_{\max}\right), W_i\left(t_{\max}-1\right)\right) =$$
$$= \sum_{k=1}^{n}\left|W_i\left(t_{\max}\right)(k) - W_i\left(t_{\max}-1\right)(k)\right|$$

- the mean error with respect to the eigen vectors $Er = \frac{1}{m}\sum_{i=1}^{m} E\left(W_i\left(t_{\max}\right), \Phi_i\right)$,

$$E\left(W_i\left(t_{max}\right), \Phi_i\right) = \frac{1}{n} D \sum_{k=1}^{n}\left|W_i\left(t_{max}\right)(k) - \Phi_i(k)\right|$$

The obtained results are shown in tables 1, 2 and 3.

V - GHA	V - Sanger	V- APEX	Σ	t_{max}
0.0377	0.0371	0.0491	Σ_1	75
0.0186	0.0186	0.0243	Σ_2	
0.0064	0.0054	0.0074	Σ_3	
0.0387	0.0393	0.0542	Σ_1	50
0.0276	0.0273	0.0348	Σ_2	
0.0090	0.0070	0.0102	Σ_3	
0.0896	0.0851	0.1085	Σ_1	20
0.0572	0.0499	0.0662	Σ_2	
0.0160	0.0114	0.0172	Σ_3	
0.1569	0.1751	0.1759	Σ_1	10
0.0774	0.0600	0.0858	Σ_2	
0.0202	0.0135	0.0211	Σ_3	

Table 1

Er - GHA	Er - Sanger	Er- APEX	Σ	t_{max}
0.0339	0.0423	0.0499	Σ_1	75
0.0295	0.0403	0.0426	Σ_2	
0.0379	0.0532	0.0417	Σ_3	
0.0334	0.0429	0.0485	Σ_1	50
0.0331	0.0450	0.0453	Σ_2	
0.0414	0.0561	0.0442	Σ_3	
0.0417	0.0551	0.0543	Σ_1	20
0.0434	0.0572	0.0505	Σ_2	
0.0484	0.0614	0.0493	Σ_3	
0.0555	0.0612	0.0626	Σ_1	10
0.0504	0.0629	0.0531	Σ_2	
0.0518	0.0637	0.0520	Σ_3	

Table 2

$\frac{V}{Er}$ - GHA		$\frac{V}{Er}$ - Sanger		$\frac{V}{Er}$ - APEX		Σ	t_{max}	
1.1120	↓	0.8770	↓	0.9839	↓	Σ_1	75	↑
0.6305		0.4615		0.5704		Σ_2		
0.1688		0.1015		0.1774		Σ_3		
1.1586	↓	0.9160	↓	1.1175	↓	Σ_1	50	
0.8338		0.6066		0.7682		Σ_2		
0.2173		0.1247		0.2307		Σ_3		
2.1486	↓	1.5444	↓	1.9981	↓	Σ_1	20	
1.3179		0.8723		1.3108		Σ_2		
0.3305		0.1856		0.3488		Σ_3		
2.8270	↓	2.8611	↓	2.8099	↓	Σ_1	10	
1.2305		0.9538		1.6158		Σ_2		
0.3171		0.2119		0.4057		Σ_3		

Table 3

According to the experimental results we formulate the following conclusions.

1. The ratio $\frac{V}{Er}$ of the stabilization coefficient *V* and the error *Er*, is fast decreasing in case of the APEX and GHA algorithms as compared to its variation in case of the Sanger variant. Also, in case of the APEX and GHA algorithms the values of the ration $\frac{V}{Er}$ remain larger then in case of Sanger algorithm, that is the APEX and GHA lead to smaller errors versus the stabilization index V.
2. The stabilization of the Sanger variant is installed faster then in case of GHA and APEX. In other words, approximates of the principal components are learned faster by the Sanger algorithm as compared to the GHA and APEX, but the performance expressed in terms of the error is lower.
3. The errors are significantly influenced by the variation of the eigen values and they are less influenced by their actual magnitude.

4 PCA Based Approach for Restoration Noisy Images

The effectiveness of restoration techniques mainly depends on the accuracy of the image modeling. Many image-degradation models have been developed based on different assumptions. One of the most popular degradation models is the linear continuous image-degradation where it is assumed that the image blur can be modeled as a superposition with an impulse response H that may be space variant and its output is subject to an additive noise.

The restoration can be viewed as a process that attempts to reconstruct or recover an image that has been degraded by using some *a priori* knowledge about the degradation phenomenon. Thus restoration techniques are oriented toward modeling the degradation and applying the inverse process in order to recover the original image. This approach usually involves formulating a criterion of goodness that will yield some optimal estimate of the desired result.

The research work reported in this paper aims to propose a new image reconstruction method based on the features extracted from the noise given by the principal components of the noise covariance matrix. The computation of the features of the noise $\eta \sim N(0,\Sigma)$ is carried out by a PCA neural network trained by the GHA as well as alternative approaches as for instance Generalized Recursive Least Square (GRLS) algorithm and APEX. The training process aims the learning of the *p* most significant eigen-vectors of Σ, that is the eigenvectors corresponding to the largest *p* eigen-values of Σ.

We consider the additive normal distributed degradation model. Let I^0 be a RxC matrix, where $C = nC_1, 2 \le n < C$ representing the initial image of *L* gray levels and let *I* be the distorted variant resulted from I^0 by superimposing random noise $N(0,\Sigma)$,

$\forall i = 1,...,R$, $k = n(j-1),...,nj$, $j = 1,...,C_1$, $I_{i,j}(k) = I^0_{i,j}(k) + \eta(k)$, where

- $I_{i,j}$ is the sequence of *n* pixels of the *i-th* row from the $n(j-1)$-th pixel to the *nj*-th of the image *I*
- $I^0_{i,j}$ is the sequence of *n* pixels of the *i-th* row from the $n(j-1)$-th pixel to the *nj*-th of the image I^0
- η is a *n*-dimensional random vector distributed $N(0,\Sigma)$.

The multiresolution algorithms perform the restoration tasks by combining, at each resolution level, according to a certain rule, the pixels of a binary support image. The values of the support image pixels are either 1 or 0 depending on their significance degree. At each resolution level, the contiguous areas of the support image corresponding to 1-value pixels are taken as possible objects of the image. The multiresolution support set is a data structure suitable for developing noise removal algorithms. The multiresolution algorithms perform the restoration tasks by combining, at each resolution level, according to a certain rule, the pixels of a binary support image. The values of the support image pixels are either 1 or 0 depending on their significance degree. At each resolution level, the contiguous areas of the support image corresponding to 1-value pixels are taken as possible objects of the image. The multiresolution support is the set of all support images. The multiresolution support can be computed using the statistically significant wavelet coefficients.

The algorithm for removing the noise component proceeds in two stages:

- in the first stage the noise features Φ are computed. The columns of Φ are the eigen vectors of Σ, taken according to the decreasing order of their corresponding eigen- values;
- in the second stage, using Φ, we apply a certain noise removal method, say *M*, based on the multirezolution support set to clean each pixel (i,j) of the decorrelated transformed image.

The restoration process of the image *I* using the learned features is performed as follows:

Step 1. Compute the image *I'* by decorrelating the noise component,

$\forall i = 1,...,R$, $j = 1,...,C_1$,

$I'_{i,j} = \Phi^T I_{i,j} = \Phi^T I^0_{i,j} + \eta'$, where

$\eta' = \Phi^T \eta \sim N(0,\Sigma'), \Sigma' = \Phi^T \Sigma \Phi = \Lambda$,

$\Lambda = diag\{\lambda_1, \lambda_2, ..., \lambda_n\}$.

Step 2. The noise component η' is removed for each pixel *P* of the image *I'* using the multirezolution support of *I'* by the labeling method of each wavelet coefficient of P, resulting *I"*.

$I''_{i,j} = M(I'_{i,j}) \cong \Phi^T I^0_{i,j}, \forall i = 1,...,R$, $j = 1,...,C_1$

where $M(I'_{i,j})$ is produced by applying the above mentioned method to $I'_{i,j}$.

Step 3. An approximation $\tilde{I} \cong I^0$ of the initial image I^0 is produced by applying the inverse transform of T_{Φ^T} to *I"* ,

$\widetilde{I}_{i,j} = \Phi I''_{i,j} \cong \Phi\Phi^T I^0_{i,j} = I^0_{i,j}$, $\forall i = 1,...,R$, $j = 1,...,C_1$

Note that the decorrelation of the noise component is performed by the computation carried out at **Step 1** because the resulted image is

$I'_{i,j}(k) = \Phi^T I^0_{i,j}(k) + \eta'(k)$, $\quad k = n(j-1),...,nj$, $j = 1,...,C_1$,

where for each $k = n(j-1),...,nj$, $j = 1,...,C_1$, $\eta'(k) \sim N\left(0, \sigma^2_{i,k}\right)$, $\sigma^2_{i,k} = \lambda_k$.

Our tests were performed on monochrome images distorted by $N(m, \sigma^2)$ - distributed noise. The learning process encoded the principal components in about 2000 steps, the quality of the restored image being “enough good” for the approximations resulted so far. In the sequel, we present some samples represented in figures 1 and 2, where $m = 50$ and $\sigma^2 = \max_k \lambda_k = 100$.

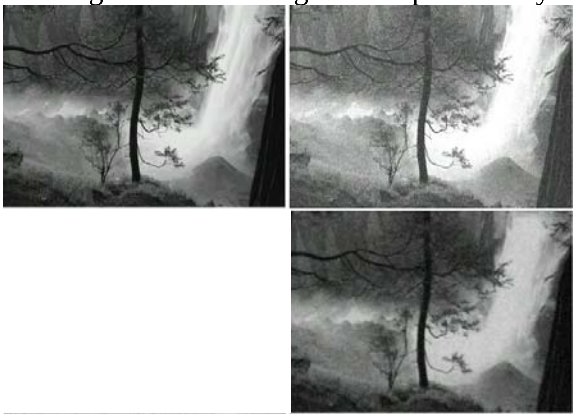
5 PCA Based Shrinkage Technique for Noise Removal

Image restoration is very important in remote sensing where aerial photographs have to be registered against the map, or to aerial photographs of the same region have to be registered with each other.

Image restoration methods are used to improve the appearance of an image by application of a restoration process that uses a mathematical model for image degradation. Examples of the types of degradation include blurring caused by motion or atmospheric disturbance, geometric distortion caused by imperfect lenses, superimposed interference patterns caused by mechanical systems, and noise from electronic sources.

Usually, it is assumed that the degradation model is either known or can be estimated. The general idea is to model the degradation process and then apply the inverse process to restore the original image. In practice, the degradation process model is often not known and must be experimentally determined or estimated. Any available information regarding the images and the systems used to acquire and process them proves helpful. These information combined with the developer’s experience, can be applied to solve the specific application. Knowledge of the image creation process is application specific; for example, it is helpful to know how a specific lens distorts an image or how mechanical vibration from a satellite affects an image. This information may be provided by knowledge about the image acquisition process itself, or it may be extracted from the degraded images by applying image analysis techniques.

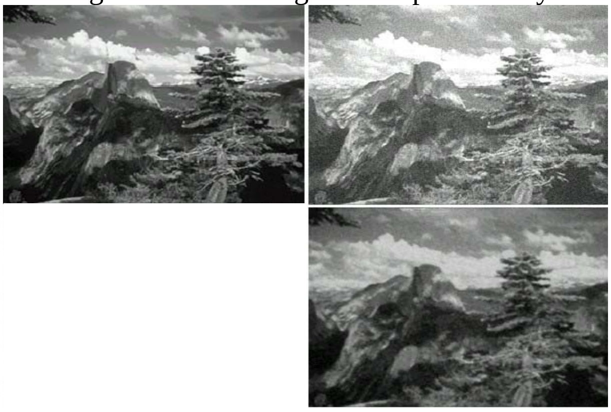
The original “clean” image A sample of noisy variant



The restored image

Fig. 1

The original “clean” image A sample of noisy variant



The restored image

Fig. 2

Noise is any undesired information that contaminates an image and appears in images from a variety of sources. The digital image acquisition process, which converts an optical image into a continuous electrical signal that is then sampled, is the primary process by which noise appears in digital images. At every step in the process, there are fluctuations caused by natural phenomena that add a random value to the exact brightness value for a given pixel. Typically, the noise can be modeled with either a Gaussian, uniform or salt and pepper distribution.

The image restoration tasks mainly correspond to the process of finding an approximation to the overall degradation process and finding the appropriate inverse process to estimate the original unknown image.

Let *x* be the original nongaussian random variable, η be a Gaussian noise of zero mean and variance σ^2 and $y = x + \eta$ the observable random variable. Denoting by *p* the probability density of *x* and by $f = -\log p$, the maximum likelihood method gives the following estimation for *x*,

$$\hat{x} = \arg\min_u \frac{1}{2\sigma^2}(y-u)^2 + f(u). \qquad (5.1)$$

If we assume that *f* is strictly convex and differentiable, the maximum likelihood (ML) estimator $\hat{x}$ is given by,

$$\frac{1}{\sigma^2}(\hat{x}-y) + f'(x) = 0. \qquad (5.2)$$

We obtain $\hat{x} = g(y)$, where

$$g^{-1}(u) = u + \sigma^2 f'(u).$$

In the general case, even if *g* cannot be inverted, the following first-order approximation of the ML estimator with respect to noise level is $\hat{x}' = y - \sigma^2 f'(y)$.

Note that the sign of the estimator $\hat{x}'$ is often different from the sign of *y* even for symmetrical zero-mean densities. In order to alleviate this problem, one may use the estimator [10],

$$\hat{x}_0 = \operatorname{sign}(y)\max\left(0, |y| - \sigma^2 |f'(y)|\right). \qquad (5.3)$$

In order to use the estimator $\hat{x}_0$ in practice, we consider the classical Laplace density, $p(x) = \frac{1}{\sqrt{2}}\exp\left(-\sqrt{2}|x|\right)$.

The ML denoising nonlinearity *g* takes the form,

$$g(u) = \operatorname{sign}(u)\max\left(0, |u| - \sqrt{2}\sigma^2\right). \qquad (5.4)$$

The function corresponding to Laplace density is a shrinkage function that reduce the absolute value of its argument by a fixed amount [10].

The noise removal technique is based on a transform that decorrelates the components of the noise followed by the use of a shrinkage process in order to remove the noise.

The experimental results are performed on data represented by monochrome images decomposed in blocks of 8×8 size and liniarization. The data are preprocessed in order to get a normalized representation and centering process. Each image is represented as,

$$\mathbf{X} = \mathbf{X}_0 + \boldsymbol{\eta},$$

where $\mathbf{X}_0$ is the original image, and $\boldsymbol{\eta}$ is the noise component. The working assumptions are that both of the repartition of $\mathbf{X}_0$ and $\boldsymbol{\eta}$ are unknown, but the statistical information represented by the mean vectors and the covariance matrices is available. Also, we assume that $\mathbf{X}_0$ and $\boldsymbol{\eta}$ are uncorrelated.

The aim is to obtain an approximation of $\mathbf{X}_0$ from the input **X** on the basis of $\boldsymbol{\mu}_\eta = E(\boldsymbol{\eta})$, $\boldsymbol{\Sigma}_\eta = Cov\left(\boldsymbol{\eta}, \boldsymbol{\eta}^T\right)$, $\boldsymbol{\mu}_0 = E(\mathbf{X}_0)$ and $\boldsymbol{\Sigma}_0 = Cov\left(\mathbf{X}_0, \mathbf{X}_0{}^T\right)$.

Let **Y** be the centered data,

$$\mathbf{Y} = \mathbf{X} - \mathrm{E}(\mathbf{X}) = \mathbf{X}_0 - \boldsymbol{\mu}_0 + \boldsymbol{\eta} - \boldsymbol{\mu}_\eta. \qquad (5.5)$$

Then,

$$\mathrm{Cov}\left(\mathbf{Y}, \mathbf{Y}^T\right) = \boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_\eta. \qquad (5.6)$$

We denote by n the dimension of input data. Let **A** be the matrix having as columns the eigen vectors of $\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}_\eta$ and $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$, where $\lambda_1, \lambda_2, ..., \lambda_n$ are the corresponding eigen values. It is well known that although $\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}_\eta$ is not symmetric, $\lambda_1, \lambda_2, ..., \lambda_n$ are positive real numbers.

$$\mathbf{A}^T \boldsymbol{\Sigma}_0 \mathbf{A} = \mathbf{I}_n \qquad (5.7)$$

$$\mathbf{A}^T \boldsymbol{\Sigma}_\eta \mathbf{A} = \boldsymbol{\Lambda} \qquad (5.8)$$

Then the covariance matrix of $\mathbf{Z} = \mathbf{A}^T \mathbf{Y}$ is,

$$\mathrm{Cov}\left(\mathbf{Z}, \mathbf{Z}^T\right) = \mathbf{A}^T\left(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_\eta\right)\mathbf{A} = \mathbf{I}_n + \boldsymbol{\Lambda} \qquad (5.9)$$

that is the components of Z are uncorrelated. Since

$$\mathbf{Z} = \mathbf{A}^T \mathbf{Y} = \mathbf{A}^T\left(\mathbf{X}_0 - \boldsymbol{\mu}_0\right) + \mathbf{A}^T\left(\boldsymbol{\eta} - \boldsymbol{\mu}_\eta\right), \qquad (5.10)$$

the component $\boldsymbol{\eta}' = \mathbf{A}^T\left(\boldsymbol{\eta} - \boldsymbol{\mu}_\eta\right)$ is the noise affecting the transformed input data, $\mathbf{A}^T\left(\mathbf{X}_0 - \boldsymbol{\mu}_0\right)$.

The covariance matrix of $\boldsymbol{\eta}'$ is,

$$\mathrm{Cov}\left(\boldsymbol{\eta}', \boldsymbol{\eta}'^{T}\right) = \mathbf{A}^{T} Cov\left(\boldsymbol{\eta} - \boldsymbol{\mu}_{\eta}, \left(\boldsymbol{\eta} - \boldsymbol{\mu}_{\eta}\right)^{T}\right)\mathbf{A} =$$

$$= \mathbf{A}^{T} \boldsymbol{\Sigma}_{\eta} \mathbf{A} = \boldsymbol{\Lambda}$$

Using the shrinkage functions

$$g(u) = \mathrm{sign}(u)\max\left(0, |u| - \sqrt{2}\lambda_{i}\right), \qquad (5.11)$$

applied to the pixels of the image **Z**, we get $\mathbf{Z_0}$ a good approximation of the noise free version of **Z**.

The restored version $\mathbf{X_0}$ is then obtained by applying the inverse transforms,

$$\mathbf{Z}_{0} = \mathbf{A}^{T}\mathbf{Y}_{0},$$

$$\mathbf{X}_{0} = \mathbf{Y}_{0} + \mathrm{E}(\mathbf{X})$$

Using that $\mathbf{A}\mathbf{A}^{T} = \boldsymbol{\Sigma}_{0}^{-1}$, we get

$$\mathbf{A}\mathbf{Z}_{0} = \boldsymbol{\Sigma}_{0}^{-1}\mathbf{Y}_{0}\text{, where} \qquad (5.12)$$

$$\mathbf{Y}_{0} = \boldsymbol{\Sigma}_{0}\mathbf{A}\mathbf{Z}_{0}. \qquad (5.13)$$

Note that in case the mean of noise is 0, the mean of $\mathbf{X_0}$ is obviously $\boldsymbol{\mu_0}$ and by applying the described method yields to reduce the noise variance [9],[10]. When the mean of the noise is non zero, the mean also influences the still remaining noise and, consequently, the performance of the described method is less than in the previous case.

The small values of the eigen values are mainly due to the noise and consequently they and their corresponding eigen vectors are neglected. We performed several tests on the proposed algorithm aiming to derive conclusions concerning the time efficiency and the quality of the resulted restored images.

The tests were performed on gray level images decomposed in blocks of size 8x8. The data were processed to get normalized centered variants. The samples of data included images sharing similar statistical properties as well as very dissimilar images.

The tests proved that the proposed restoration technique yields high quality restored images. The results of a test of this alghoritm in case of 0 mean Gaussian noise are depicted in figure 3 and figure 4.

A comparative analysis of the performance of this algorithm has been developed against most currently used noise removal algorithms. In our opinion, the results are quite promising and several possible extensions of it are still in progress.

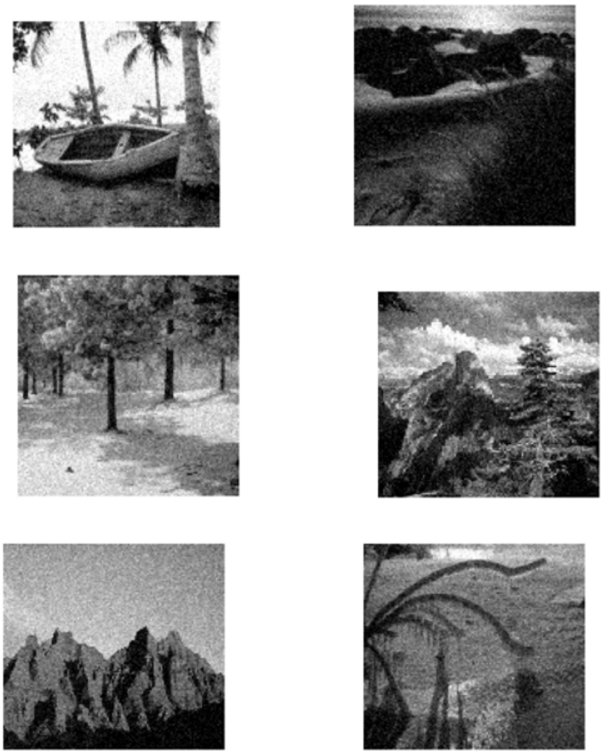


Fig. 3: A set of noisy samples

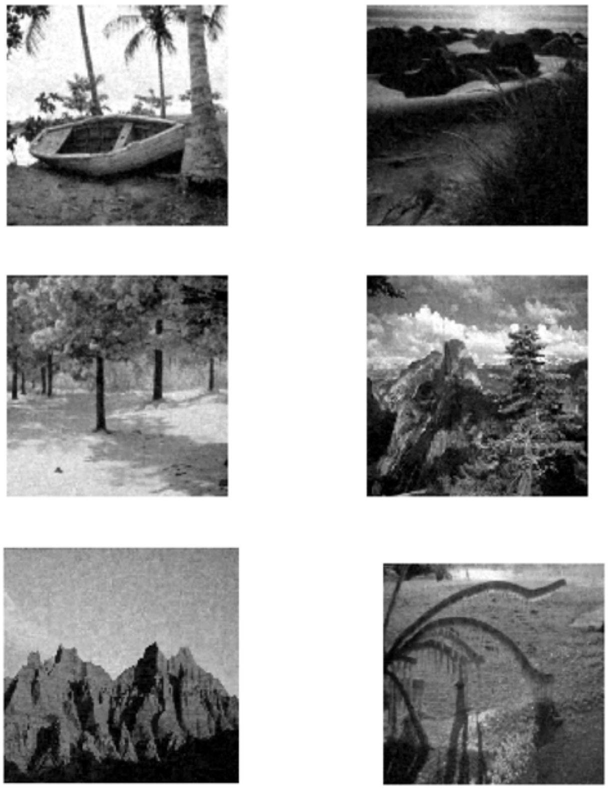


Fig. 4. The results obtained using the proposed algorithm

References:

[1] F. Araddiga, R. Donat, J.C. Trilloo, Image compression using the SVD Algorithm combined with Harten's multiresolution, In: *WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICA TIONS*, Issue 9, Volume 2, 2005

[2] C. Chatterjee, V.P. Roychowdhury, E.K.P. Chong, On Relative Convergence Properties of Principal Component Analysis Algorithms, *IEEE Transaction on Neural Networks*, vol.9,no.2, 1998

[3] Cocianu, C., State, L., Vlamos, P.,2002, On a Certain Class of Algorithms for Noise Removal in Image Processing:A Comparative Study, In *Third IEEE Conference on Information Technology ITCC-2002, Las Vegas,* Nevada, USA, April 8-10, 2002

[4] Cocianu, C., State, L., Stefanescu, V., Vlamos, P., 2004, On the Efficiency of a Certain Class of Noise Removal Algorithms in Solving Image Processing Tasks, In: *Proceedings of the ICINCO 2004*, Setubal, Portugal

[5] K.I. Diamantaras, S.Y. Kung, *Principal Component Neural Networks: theory and applications* , John Wiley &Sons,1996

[6] J. Karhunen, E. Oja, New Methods for Stochastic Approximations of Truncated Karhunen-Loeve Expansions, *Proceedings 6th International Conference on Pattern Recognition*, Springer Verlag, 1982

[7] S.Y. Kung, K.I. Diamantaras, A Neural Network Learning Algorithm for Adaptive Principal Component Extraction (APEX), *IEEE Transaction on Acoustics, Speech and Signal Processing*, 1990

[8] H.J. Kushner., D.S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer Verlag, 1978

[9] S. Haykin, *Neural Networks A Comprehensive Foundation*, Prentice Hall,Inc. 1999

[10] A. Hyvarinen, J. Karhunen, E. Oja, *Independent Component Analysis*, John Wiley &Sons, 2001

[11] A. Hyvarinen, P. Hoyer, P., E. Oja, Image Denoising by Sparse Code Shrinkage, www.cis.hut.fi/projects/ica, 1999

[12] J. Mao, A.K. Jain, Artificial Neural Networks for Feature Extraction and Multivariate Data Projection, *IEEE Transaction on Neural Networks*, vol.6, no.2, 1995

[13] K. Matsuoka, M. Kawamoto, A Neural Network that Self-Organizes to Perform Three Operations Related to Principal Component Analysis, *Neural Networks*, vol.7, no.5, 1994

[14] E. Oja, *Data Compression, Feature Extraction and Autoassociation in Feedforward Neural Networks*, vol. 1, North Holland, 1991

[15] E. Oja, "Principal Components, Minor Components and Linear Neural Networks", *Neural Networks*, vol. 5, 1992

[16] T.D. Sanger, An Optimality Principle for Unsupervised Learning, *Advances in Neural Information Systems*, ed. D.S. Touretzky, Morgan Kaufmann, 1989

[17] L. State, C. Cocianu, P. Vlamos, Attempts in Using Statistical Tools for Image Restoration Purposes, *Proceedings of SCI2001*, Orlando, USA, July 22-25, 2001.

[18] L. State, C. Cocianu, V. Stefanescu, P. Vlamos, "PCA-Based Data Mining Probabilistic and Fuzzy Approaches with Applications in Pattern Recognition", *Proceedings of ICSOFT 2006*, Portugal, pp. 55-60, 2006

[19] A. Tarczynski, Reliability of Signal Reconstruction from Arbitrarily Distributed Noisy Samples, In: *WSEAS Transactions on Signal Processing*, issue 7, vol. 2, July 2006, pp. 925 – 932.

[20] F. Wenzel, R. R. Grigat, Design Aspects for the Rapid Development of Image Processing Algorithms *WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICA TIONS*, Issue 9, Volume 2, 2005